AD-A197 940

4

AAMRL-TR-88-007

ARTICULATED TOTAL BODY MODEL ENHANCEMENTS, Volume 3: Programmer's Guide

Louise A. Obergefell
Biodynamics and Bioengineering Division
Harry G. Armstrong Aerospace Medical Research Laboratory

Ints Kaleps, PhD
Biodynamics & Bioengineering Division
Harry G. Armstrong Aerospace Medical Research Laboratory

Thomas R. Gardner
System Research Laboratories, Inc.
2800 Indian Ripple Road
Dayton, Ohio 45440

John T. Fleck
J & J Technologies, Inc.
92 Henning Drive
Orchard Park, New York 14127

February 1988

Final Report

Approved for public release; distribution is unlimited.

HARRY G. ARMSTRONG AEROSPACE MEDICAL RESEARCH LABORATORY
HUMAN SYSTEMS DIVISION
AIR FORCE SYSTEMS COMMAND
WRIGHT-PATTERSON AIR FORCE BASE, OHIO 45433-6573

NOTICES

When US Government drawings, specifications, or other data are used for any purpose other than a definitely related Government procurement operation, the Government thereby incurs no responsibility nor any obligation whatsoever, and the fact that the Government may have formulated, furnished, or in any way supplied the said drawings, specifications, or other data, is not to be regarded by implication or otherwise, as in any manner licensing the holder or any other person or corporation, or conveying any rights or permission to manufacture, use, or sell any patented invention that may in any way be related thereto.

Please do not request copies of this report from the Armstrong Aerospace Medical Research Laboratory. Additional copies may be purchased from:

National Technical Information Service
5285 Port Royal Road
Springfield, Virginia 22161

Federal Government agencies and their contractors registered with Defense Technical Information Center should direct requests for copies of this report to:

Defense Technical Information Center
Cameron Station
Alexandria, Virginia 22314

TECHNICAL REVIEW AND APPROVAL

AAMRL-TR-88-007

This report has been reviewed by the Office of Public Affairs (PA) and is releasable to the National Technical Information Service (NTIS). At NTIS, it will be available to the general public, including foreign nations.

This technical report has been reviewed and is approved for publication.

FOR THE COMMANDER

Henning E. von Gierke

HENNING E. VON GIERKE, Dr Ing
Director
Biodynamics and Bioengineering Division
Armstrong Aerospace Medical Research Laboratory

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1a. REPORT SECURITY CLASSIFICATION Unclassified	1b. RESTRICTIVE MARKINGS
2a. SECURITY CLASSIFICATION AUTHORITY	3. DISTRIBUTION/AVAILABILITY OF REPORT Approved for public release; distribution unlimited.
2b. DECLASSIFICATION / DOWNGRADING SCHEDULE	
4. PERFORMING ORGANIZATION REPORT NUMBER(S) AAMRL-TR-88-007	5. MONITORING ORGANIZATION REPORT NUMBER(S)
6a. NAME OF PERFORMING ORGANIZATION Harry G. Armstrong Aerospace Medical Research Laboratory 6b. OFFICE SYMBOL (If applicable) AAMRL/BBM	7a. NAME OF MONITORING ORGANIZATION
6c. ADDRESS (City, State, and ZIP Code) Wright-Patterson AFB OH 45433-6573	7b. ADDRESS (City, State, and ZIP Code)
8a. NAME OF FUNDING/SPONSORING ORGANIZATION 8b. OFFICE SYMBOL (If applicable)	9. PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER
8c. ADDRESS (City, State, and ZIP Code)	10. SOURCE OF FUNDING NUMBERS

PROGRAM ELEMENT NO.	PROJECT NO.	TASK NO.	WORK UNIT ACCESSION NO.
62202F	7231	23	01

11. TITLE (Include Security Classification)
ARTICULATED TOTAL BODY MODEL ENHANCEMENTS
VOLUME 3: PROGRAMMER'S GUIDE

12. PERSONAL AUTHOR(S) Obergefell, Louise A., Kaleps, Ints, AAMRL; Gardner, Thomas R., Systems Research Laboratories, Inc.; Fleck, John T., J&J Technologies, Inc.

13a. TYPE OF REPORT	13b. TIME COVERED	14. DATE OF REPORT (Year, Month, Day)	15. PAGE COUNT
Final	FROM 83 Nov TO 87 Dec	88 JAN	443

16. SUPPLEMENTARY NOTATION
Effort partially funded by the National Highway Traffic Safety Administration

17. COSATI CODES

FIELD	GROUP	SUB-GROUP
20	11	
12	05	

18. SUBJECT TERMS (Continue on reverse if necessary and identify by block number)
Computer Simulation; Rigid Body Dynamics; Mathematical Model; Three-Dimensional Dynamics; Articulated Total Body Model, Crash Victim Simulator;

19. ABSTRACT (Continue on reverse if necessary and identify by block number)

The Articulated Total Body (ATB) Model is used at the Armstrong Aerospace Medical Research Laboratory (AAMRL) to study human body biomechanics in various dynamic environments, especially aircraft ejection with windblast exposure. In order to improve the model's predicted results and capabilities, a number of modifications have been made. These modifications include the capability to have segment contact ellipsoids block the wind from other segments, an option to prescribe velocity dependent wind forces, a correction to prevent angular drift in the joints, improved contact force calculations for segment contact near a plane's edge, the capability to specify as input multi-axis angular displacements to describe the vehicle motion, a sliding joint capability and a hyperellipsoid option. Along with these major changes, a number of minor corrections and clarifications have been included to form the ATB-IV version. The results of these modifications have been documented in three volumes of which this is Volume 3, the

20. DISTRIBUTION/AVAILABILITY OF ABSTRACT ☒ UNCLASSIFIED/UNLIMITED ☐ SAME AS RPT. ☐ DTIC USERS	21. ABSTRACT SECURITY CLASSIFICATION Unclassified
22a. NAME OF RESPONSIBLE INDIVIDUAL Louise A. Obergefell	22b. TELEPHONE (Include Area Code) (513) 255-3665 22c. OFFICE SYMBOL AAMRL/BBM

DD Form 1473, JUN 86 Previous editions are obsolete.

SECURITY CLASSIFICATION OF THIS PAGE
UNCLASSIFIED

Block 19 (ABSTRACT) Continued

Programmer's Guide. It contains a complete listing of the ATB-IV program with designation of coding modifications from previous versions. Volume 1 contains a technical description of the ATB-IV modifications and Volume 2 is an updated User's Guide containing the new input description.

PREFACE

This report incorporates the work done in a number of different efforts to improve the Articulated Total Body (ATB) model's capability to simulate human body biomechanics in various dynamic environments, especially aircraft ejection with windblast exposure.

The majority of modifications to the model fall into six categories:

- wind force option
- joint drift correction
- edge effect option
- multi-axis angular displacement vehicle motion prescription
- slip joint option
- hyperellipsoid option

These improvements have been combined to form the ATB-IV version on the Armstrong Aerospace Medical Research Laboratory's (AAMRL) Concurrent computer system at Wright Patterson Air Force Base. AAMRL, Systems Research Laboratories, Inc. and J&J Technologies Inc. and the National Highway Traffic Safety Administration have all contributed to the technical work described herein.

TABLE OF CONTENTS

1.0 INTRODUCTION

The Articulated Total Body (ATB) Model is used at the Armstrong Aerospace Medical Research Laboratory (AAMRL) for predicting gross human body response in various dynamic environments, especially aircraft ejection with windblast exposure. Aerodynamic force application and a harness belt capability were added to the Crash Victim Simulation (CVS) Program (Ref 1) by Calspan Corporation in 1975 for AMRL (Ref 2), and the resulting program became known as the ATB model. In 1980, Calspan made a number of modifications to the ATB model combining it with the then current 3-D Crash Victim Simulation program to form the ATB-II model (Ref 3). Complete documentation of the ATB-II program was performed by Calspan Corp. (Ref 4). A new version, ATB-III, was generated which included the improvements made by J&J Technologies Inc to model the body response to windblast for AMRL (Ref 5).

A number of efforts have been made to improve various aspects of the ATB-III model, with emphasis on its capability to simulate aircraft ejection with windblast exposure and complex automobile accidents.

This volume, Programmer's Guide, contains a complete listing of the ATB-IV.0 source code and other information about the FORTRAN program. Much of this volume is a reprinting with modifications and updates of Volume 4 of Calspan's report on the CVS (Ref 4).

A list of the variables contained in the labeled COMMON blocks and a brief description of each variable are in section Two of this volume. Cross reference charts for the subroutines and COMMON blocks are in Section Three, while a list of all the subroutines is in Section Four. Section Five contains a complete listing of the ATB-IV.0 source code.

2.0 COMMON BLOCK VARIABLES

This section contains a list of all of the variables contained in the labeled COMMON blocks of the ATB program. They are listed in the alphabetical order of the COMMON block names. Following each variable is its dimension, if any, and a short definition. If the variable is supplied as ATB program input, references are indicated to the input card number and a more complete definition may be found in the input description contained in Volume 2.

COMMON /AB DATA/

ZDEP	(3,5)	Deployment point of airbag in local reference of 1st reaction panel (Card D.4.c)
DBR	(3,3,5)	Direction cosine matrix of airbag relative to vehicle
DPVCTR	(3,5)	Vector along which airbag c.g. lies during bag inflation
DEPLOY	(3,5)	Location of deployment point
AB	(3,5)	Semiaxes of fully inflated ellipsoid airbag (Cards D.4.b)
B	(9,4,5)	3 X 3 matrix defining ellipsoid $X^TBX=1$ for reaction panel
ZR	(3,4,5)	Location of panel c.g. in vehicle reference (Card D.4.h)
BFB	(3,4,5)	c.g. offset of reaction panel (Card D.4.g)
DRR	(9,4,5)	Direction cosine matrix of reaction panel relative to inertial reference
VBAGG	(5)	Geometric volume of fully inflated airbag
VSCS	(5)	Coefficient of sliding friction of the airbag (Card D.4.f)
SPRK	(5)	Spring constant of a linear spring used to stipulate attachment of the airbag at the deployment point (Card D.4.f)

CK	(5)	Parameter used to stabilize airbag numerical integration (Card D.4.f)
CMASS	(5)	Multiplier to increase or decrease the mass of the airbag to artificially dampen the integrated airbag motion (Card D.4.f)
CYMIN	(5)	Mass flow into the airbag
CYMOUT	(5)	Mass flow out of the airbag
BAGPV	(5)	Undeformed airbag volume
PD	(5)	Airbag pressure differential
VBAG	(5)	Airbag volume
VOLBP	(5)	Total volume of intersection or airbag with contacting segments and panels
PCYV	(5)	Volume of mass flow into airbag at atmospheric pressure at time if initial inflation
PCYMIN	(5)	Mass flow into airbag at time of initial full inflation
PVBAG	(5)	Airbag volume ~t time if initial inflation
TV1	(3,4,5)	Memory for Subroutines INTERS and EDEPTH for airbag-panel ellipsoid contacts
TV2	(3,10,5)	Memory for Subroutines INTERS and EDEPTH for airbag-segment ellipsoid contacts
SWITCH	(5)	Reciprocal density of airbag at time of initial full inflation

PYMOUT	(5)	Mass flow out of airbag at time of initial full inflatiou
SCALE	(5)	Ratio (0-1) of linear dimensions of airbag to fully inflated airbag
PREVT		Value of TIME at previous airbag integration step
IFULL	(6)	Indicates that airbag is fully inflated

<u>COMMON</u> <u>/CDINT/</u>

UU	(4)	Constants computed by Subroutine TRIGFS valid for the upcoming integrator time step
GH	(3,4)	
E	(3,240)	Intermediate storage of function evaluations in Subroutine DINT
FF	(5,240)	
GG	(5,240)	
Y	(5,240)	
U	(5,240)	
H		Current value of the independent variable step size in Subroutine DINT

HPRINT		Saved value of H while integrating to print point only
TSAVE		Set to zero or H to reset integrator
TPRINT		Value of next print time point
TSTART		Start time of an integration step
ICNT		Count of successive integration steps for which convergence has been successful
IDBL		Maximum value for ICNT before test to double step size is performed
IFLAG		Currently not used by program

COMMON /CEULER/

IEULER	(30)	Current lock-unlock conditions for an Euler joint (see identification under IPIN on Cards B.2)
HIR	(3,3,90)	Direction cosine matrix defining orientation of axes of an Euler joint
ANG	(3,30)	Angles of orientation of an Euler joint
ANGD	(3,30)	Time derivative of orientation angles of an Euler joint
FE	(3,30)	Components of torque acting on an Euler joint in joint reference

TQE	(3,30)	Components of torque acting on an Euler joint in inertial reference
CONST	(5,30)	Memory of previous angles of orientation of an Euler joint

COMMON /CMATRX/

V1	(3,30)	Right hand side of system of equations $B_{11}\ddot{x} + B_{12}\dot{w} + B_{13} f = V_1$
V2	(3,30)	Right hand side of system of equations $B_{22}\ddot{x} + B_{24} t = v_2$
V3	(3,12)	Right hand side of system of equations $B_{31} \ddot{x} + B_{32}\dot{w} + B_{35} q = V_3$
B12	(3,3,60)	Subarray elements of B_{12}
A22	(3,3,60)	Subarray elements of A_{22}
F	(3,30)	Components of force acting on the joints from the solution of system equations
TQ	(3,30)	Components of torque acting on the joints from the solution of system equations
WJ	(30)	Relative angular velocity of each joint
A11	(3,3,30)	Subarray elements of A_{11}

COMMON /CNSNTS/

PI		FORTRAN Subroutine Library value for Pi, computed by PI = DATAN2(0.0D0,-1.0D0)
RADIAN		Number of radians per degree (PI/180)
G		Resultant of gravity vector (Card A.3)
THIRD		Double precision value for 1/3
EPS	(24)	Values of negative powers of ten, computed by EPS(I) = 10.0D0**(-I)
UNITL		I/O unit of length (Card A.3)
UNITM		I/O unit of force of mass (Card A.3)
UNITT		I/O unit of time (Card A.3)
GRAVITY	(3)	Components of gravity vector (Card A.3)
TWOPI		2.0*PI

COMMON /CNTSRF/

PL	(24,30)	Array of parameters that define each plane (See Table 1 in Volume 1)
BELT	(20,8)	Array of parameters that define each belt (Cards D.3.b-D.3.c)
TPTS	(6,8)	Location of belt tangent points in inertial reference

BD	(24,40)	Array of parameters that define each ellipsoid

COMMON /COMAIN/

VAR	(240)	Integrated function values supplied by Subroutine DINT to Subroutine PDAUX
DER	(240)	Function derivatives supplied by Subroutine PDAUX to Subroutine DINT
DT		Time interval for main program output time points (Card A.4)
HO		Initial integrator step size (Card A.4)
HMAX		Maximum integrator step size (Card A.4)
HMIN		Minimum integrator step size (Card A.4)
RSTIME		Restart time (Card A.1.a)
ISTEP		Current integration step number
NSTEPS		No. of integration steps for duration of simulation (Card A.4)
NDINT		No. of iterations for convergence test for Subroutine DINT (Card A.4)
NEQ		Total number of functions integrated by Subroutine DINT

IRSIN	Restart input unit no. (Card A.1.a)
IRSOUT	Restart output unit no. (Card A.1.a)

COMMON /CONTRL/

TIME	Current simulation time
NSEG	Number of body segments of crash victim, max=30 (Card B.1)
NJNT	Number of joints, max=30 (Card B.1)
NPL	Number of plane definitions supplied on Cards D.2, max=30 (Card D.1)
NBLT	Number of belt definitions supplied on Cards D.3, max=8 (Card D.1)
NBAG	Number of airbag definitions supplied Cards D.4, max=5 (Card D.1)
NVEH	Segment identification number for the vehicle (NVEH=NSEG+number of vehicles)
NGRND	Segment identification number for the ground (NGRND=NSEG+NBAG+number of vehicles+1)
NS	Number of singular segments, i.e., W or at least one component of PHI is zero
NQ	Number of contraints supplied on Cards D.6, final max = 12 (Card D.1)

NSD		Number of spring dampers supplied on Cards D.8, max=20 (Card D.1)
NFLX		Total number of interior segments of all flexible elements.
NHRNSS		Number of harness-belt systems supplied on Cards F.8, max=5 (Card D.1)
NWINDF		Number of wind force functions supplied on Cards E.6 (Card D.1)
NJNTF		Number of joint restoring force functions supplied on Cards E.7 (Card D.1)
NPRT	(36)	Indicators that control optional output of the program (Card A.5)
NPG		Current page number of main output

COMMON /CSTRNT/

A13	(3,3,24)	Subarray elements of A_{13} for system of equations $M\ddot{x} + A_{11}f + A_{13}q - U_1$
A23	(3,3,24)	Subarray elements of A_{23} for system of equations $\phi\dot{w} + A_{21}f + A_{22}t + A_{23}q = U_2$
B31	(3,3,24)	Subarray elements of B_{31} for system of equations defining constraints

B32	(3,3,24)	Subarray elements of B_{32} for system of equations defining constraints
HHT	(3,3,12)	Array hh^T or $I-hh^T$ for each constraint
RK1	(3,12)	Specified point on segment number KQ1 (Card D.6)
RK2	(3,12)	Specified point on segment number KQ2 (Card D.6)
QQ	(3,12)	Computed force necessary to maintain each constraint
TQQ	(3,12)	Normal vector at the point of contact for each constraint
RQQ	(3,12)	R dot term for constraint equation
HQQ	(3,12)	Reference vector at point of constraint
SQQ	(3,12)	R term for constraint equation
CFQQ	(12)	Coefficient of friction for each constraint
KQ1	(12)	Segment identification number of the 1st specified point (Card D.6)
KQ2	(12)	Segment identification number of the 2nd specified point (Card D.6)
KQTYPE	(12)	Constraint type number (Card D.6)

COMMON /CYDATA/

CYTD	(5)	Gas supply actuator firing time (Card D.4.d)
CYPA	(5)	Atmospheric pressure (Card D.4.d)
CYSP	(5)	Initial gas supply pressure (Card D.4.d)
CYTO	(5)	Initial gas supply temperature (Card D.4.d)
CYVO	(5)	Gas supply reservoir volume (Card D.4.d)
CYCD	(5)	Sonic throat discharge coefficient (Card D.4.e)
CYK	(5)	Ratio of specific heats of supply gas (Card D.4.e)
CYR	(5)	Specific gas constant (Card D.4.e)
CYAT	(5)	Sonic throat area (Card D.4.e)
CYPV	(5)	Vent pressure of the exhaust orifice
CYCDO	(5)	Exhaust orifice discharge coefficient (Card D.4.e.)
CYAO	(5)	Exhaust orifice area (Card D.4.f)
CYPO	(5)	Initial air cylinder gauge supply pressure
CYSS	(5)	Speed of sound
CYLO	(5)	Characteristic length

CYC	(5)	Air cylinder gas constant
CYRHOO	(5)	Initial air cylinder density
CYVMAX	(5)	Air cylinder maximum volume
CYORFC	(5)	Air cylinder exhaust orifice constant
CYRHO	(5)	Density of air cylinder gas supply
CYT	(5)	Temperature of air cylinder gas supply
CYP	(5)	Pressure of air cylinder gas supply
CYV	(5)	Volume of air cylinder gas supply at standard atmospheric pressure

COMMON /DAMPER/

APSDM	(3,20)	Attachment point in local reference of segment M for spring dampers (Card D.8)
APSDN	(3,20)	Attachment point in local reference of segment N for spring dampers (Card D.8)
ASD	(5,20)	Spring and viscous force function coefficients (Card D.8)
MSDM	(20)	Identification number of segment M (Card D.8)
MSDN	(20)	Identification number of segment N (Card D.8)

COMMON /DESCRP/

PHI	(3,30)	Segment principal moments of inertia (Cards B.2)
W	(30)	Segment weight (Cards B.2)
RW	(30)	Reciprocal mass (g/w) for each segment
SR	(4,60)	Joint locations in local reference of adjacent segments (Cards B.3)
HA	(3,60)	Principal line of joint from which flexure angle is measured
HB	(4,60)	Perpendicular to HA (pin axis if joint is pinned)
RPHI	(3,30)	Reciprocal moments of inertia for each segment
HT	(3,3,60)	Principal axes of the joints
SPRING	(5,90)	Flexural and torsional spring characteristics (Cards B.4)
VISC	(7,90)	Flexural and torsional viscous characteristics (Cards B.5)
JNT	(30)	Magnitude indicates the segment identification number that is connected to segment J+1 by joint J (Cards B.3)
IPIN	(30)	Indicator of joint type (Card B.3)

ISING	(30)	Indicator (value=1) that segment is singular
IGLOB	(30)	Input indicator (Card F.4.a) to signify that joint J is to use the globalgraphic option. A nonzero value will be set to index of function to be used.
JOINTF	(30)	The function idenfication number used to compute the joint restoring force (Card F.5)

COMMON /FLXBLE/

HF	(4,12,8)	Coefficients of quadratic function defining relative orientation of interior segments of flexible elements
B42	(3,3,24)	Subarray elements of matrix B_{42} in the constraint equations for flexible elements
V4	(3,8)	Right hand side of the constraint equations for flexible elements.
NFLEX	(3,8)	The identification numbers of reference, interior and terminating segments for each interior segment.

COMMON /FORCES/

PSF	(7,70)	Array of output values for plane-segment contacts
BSF	(4,20)	Array of output values for belt-segment contacts
SSF	(10,40)	Array of output values for segment-segment contacts
BAGSF	(3,20)	Array of output values for airbag-segment contacts
PRJNT	(7,30)	Output arrays for joint parameters
NPANEL	(5)	Number of reaction panels for each airbag (J=1, NBAG)
NPSF		Number of plane-segment contact (Max=70)
NBSF		Number of belt-segment contacts (Max=20)
NSSF		Number of segment-segment contacts (Max=40)
NBGSF		Number of items to be printed for airbag-segment contacts (Max=20)

COMMON /HRNESS/

BAR	(15,100)	Coordinates of points in local reference (Cards F.8.d)
BB	(100)	Lengths of individual belt segments between reference points
BBDOT	(100)	Time derivative of belt segment lengths
PLOSS	(2,100)	Energy loss of individual belt segments
XLONG	(20)	The initial slack of each belt (Cards F.8.c)
HTIME	(2)	Previous value of TIME for Subroutine HPTURB
IBAR	(5,100)	Array of indicators containing KS, KE, NF index, NPD and NPR (Cards F.8.d) for each point
NL	(2,100)	Pointers to the IBAR and NTHRNS arrays for each point in play
NPTSPB	(20)	Number of points per belt (Cards F.8.b)
NPTPLY	(20)	Number of points in play per belt
NTHRNS	(20)	Index to NTAB array defining the force deflection functions for each belt
NBLTPH	(5)	Number of belts per harness (Card F.8.a)

COMMON /INTEST/

SGTEST	(3,4,30)	Integrator convergence test input numbers (Cards B.6)
XTEST	(3,120)	Integrator convergence test numbers setup by PDAUX for DINT
SEGT	(120)	Segment identification of integrator variable
REGT	(120)	Identification (ANG VEL, ANG ACC, LIN VEL or LIN ACC) of type of integrator variable

COMMON /JBARTZ/

MNPL	(30)	Number of segments to contact each plane (Card F.1.a)
MNBLT	(8)	Number of segments to contact each belt (Card F.2.a)
MNSEG	(30)	Number of segments to contact each segment (Card F.3.a)
MNBAG	(6)	Number of segments to contact each airbag
MPL	(3,5,30)	Segment and ellipsoid identification numbers for each plane-segment contact
MBLT	(3,5,8)	Segment and ellipsoid identification numbers for each belt-segment contact

MSEG	(3.5,30)	Segment and ellipsoid identification numbers for each segment-segment contact
MBAG	(3,10,6)	Segment and ellipsoid identification numbers for each airbag-segment contact (Cards F.4)
NTPL	(5,30)	Index to NTAB array for each plane-segment contact
NTBLT	(5,8)	Index to NTAB array for each belt-segment contact
NTSEG	(5,30)	Index to NTAB array for each segment-segment contact

<u>COMMON</u> <u>/RSAVE/</u>

XSG	(3,20,3)	Points in local segment reference for first three types of time history output (Cards H.1-H.3)
DPMI	(3,3,30)	Direction cosine matrix of principal moment of inertia to local geometric reference coordinate system for each segment
LPMI	(30)	Indicator that local geometric does not correspond to principal moment of inertia reference coordinate system for each segment (Cards B.2.i1)
NSG	(9)	Number of segments for each type of time history output (Max=20) (Cards H.1-H.9)

MSG	(20,9)	The segment identification numbers for each type of time history output (Cards H.1-H.9)
MCG		Number of bodies for body property time history output (Max=5) (Cards H.10)
MCGIN	(24,5)	Body characteristics for body property time history output (Cards H.10)
KREF	(20,9)	The reference segment numbers for each time history output (Cards H.1-H.9)

COMMON /SGMNTS/

D	(3,3,30)	Segment direction cosine matrix
WMEG	(3,30)	Segment angular velocity in local reference
WMEGD	(3,30)	Segment angular acceleration in local reference
U1	(3,30)	Total external forces on each segment
U2	(3,30)	Total external torques on each segment
SEGLP	(3,30)	Segment c.g. linear position in inertial reference
SEGLV	(3,30)	Segment c.g. linear velocity in inertial reference

SEGLA	(3,30)	Segment c.g. linear acceleration in inertial reference
NSYM	(30)	Indicators that control the symmetry options for body segments (Cards D.7)

COMMON /TABLES/

MXNTI		Dimension (50) of NTI array
MXNTB		Number of elements in the NTAB array
MXTB1		Number of elements in TAB array used to define functions
MXTB2		Total number of elements in TAB array
NTI	(50)	Index pointers to the TAB array for data defining function no. I.
NTAB	(1250)	Index pointers to TAB array for each function used for allowed contacts
TAB	(4500)	Subdivided into arrays containing function definitions and update information for each allowed contact

COMMON /TEMPVI/

CREST		Coefficient of restitution for current impulse
TTI	(3)	Value of U1 array for impulse

R1I	(3)	Value of RK1 for current constraint or impulse
R2I	(3)	Value of RK2 for current constraint or impulse
JSTOP	(4,2,30)	Indicators to signify joint is in joint stop

<u>COMMON</u> <u>/TEMPVS/</u>

Variables is this labeled common block are temporary for each subroutine that refers to it.

<u>COMMON</u> <u>/TITLES/</u>

DATE	(3)	Date of computer run in 12 alphanumeric characters (Card A.1.a)
COMENT	(40)	160 character description of the run (Cards A.1.b- A.1.c)
VPSTTL	(20)	80 character description of the crash vehicle deceleration (Card C.1)
BDYTTL	(5)	20 character description of the crash victim (Card B.1)
BLTTTL	(5,8)	20 character description of each belt (Cards D.3)
PLTTL	(5,30)	20 character description of each plane (Cards D.2)

BAGTTL	(5,6)	20 character description of each airbag (Cards D.4)
SEG	(30)	4 character segment nomenclature (Cards B.2)
JOINT	(30)	4 character joint nomenclature (Cards B.3)
CGS	(30)	1 character plot symbol of the segment C.G. (Cards B.2)
JS	(30)	1 character plot symbol of the joint location (Cards B.3)

<u>COMMON</u> <u>/VPOSTN/</u>

ZPLT	(3)	Printer plot coordinates of the vehicle reference origin (Card G.1.a)
SPLT	(3)	Scale factors for the printer plot axes (Card G.1.b)
AXV	(3,6)	Unit vector of deceleration impulse direction
VATAB	(6,501,6)	Tables of computed or supplied (Cards C.3 or C.4) values of linear (1-3) and angular accelerations (4-6) of vehicle motion
VTO	(6)	Beginning time point he deceleration table input (Card C.2)
VDT	(6)	Fixed time interval for deceleration table input (Card C.2)

TIMEV	(6)	Time duration of the deceleration impulse (Card C.2)
OMEGAV	(6)	Frequency of the half-sine wave deceleration type (Card C.2)
NVTAB	(6)	Number of points in deceleration table. Sign determines type (Card C.2)
INDXV	(6)	Segment identification number for each specified motion definition (MSEG on Card C.2.a or NVEH)

<u>COMMON</u> <u>/WINDFR/</u>

WTIME	(30)	Initial time that segment penetrates wind
QFU	(3,5)	Unit vector for force application
QFV	(3,5)	Vector for torque application
WF	(3,30)	Wind force vectors applied to segments (local)
IWIND	(30)	Indicator that wind has been penetrated
MWSEG	(7,30)	Identification numbers for the application of wind forces on each segment (Cards F.7)
NFVSEG	(6)	Segment identification number for each force function (Cards D.9)

NFVNT	(5)	Function identification number for each force function (Cards D.9)
MOWSEG	(30,30)	Contact ellipsoid numbers and segment identification numbers of blocking segments (Cards F.7)

3.0 CROSS REFERENCE CHARTS

The first two cross reference charts list the COMMON blocks used by each subroutine. The remaining ten charts list the subroutines called by each subroutine.

COMMON BLOCK CROSS REFERENCE

CALLING SUBROUTINES

```
              |MAAAAA|BBBBB|CCCCCC|DDDDDDDDD|DDDDDDDDDDDDDD|EEEEEEEE|FFFFFFG|HHHHHHHH|
              |ADIIII|EEGIL|FHIMOR|AAAAAAAAA|HIOOOORRRRSSSZ|DFJLLLUV|DILNRSL|BBEEIIPS|
              |IJRRRR|LLGNK|AANPNO|UUUUUUUUU|HNTTTTCCCIEEMP|EUOOTULA|INXTCMO|EPDRCNTE|
              |NUBBBB|TT PD|CIPUTS|XXXXXXXXX|PT33TTIQYFTTS |PNINIIRL|NPSEDSB|LLIRCPUT|
              |ASAGGG|GR UT|TNUTCS| 11233345|I 1333JUPTDQO |TCNGMLAF|IUERFOA|TANOSURC|
              | TG13G| T TA|T TET | 12212345|N   13KAR   L |HTT EBDD|TTGPLLL| YGNITB |
    ----------|------|-----|------|---------|--------------|--------|-------|--------|
    ABDATA  5 |  ****|     |      |         |              |        |       |        |
    CDINT   7 | *    |     |   *  |         | *            |        |       |    *   |
    CEULER 14 |      |   * | *    |   *     |*        *    |  *     |       |        |
    CMATRX 20 |      |     |      |*********|              |  *  *  |       |        |
C   CNSNTS 45 |***** |*****|      |*       *| *      **** *|* *  ** |  **  *|* ** ** |
A   CNTSRF 19 |  ****| * * |      |         |              |     *  |       |**   ** |
L   ----------|------|-----|------|---------|--------------|--------|-------|--------|
L   COMAIN  6 |**    |     |   *  |         | *            |        |       |        |
E   CONTRL 57 |* ****|** * | **** |*********| *       *    |* *  *  | **  * | ** *** |
D   CSTRNT 19 |      |     |      |*   *****|              |     *  | *     |        |
    CYDATA  5 |  ****|     |      |         |              |        |       |        |
C   DAMPER  7 |      |     |      |         |              |        |       |  *     |
O   DESCRP 33 |  **  |   * | *    |*********|*        *    |  *  *  | *    *|        |
M   ----------|------|-----|------|---------|--------------|--------|-------|--------|
M   FLXBLE  9 |      |   * |      |*      **|              |        |  *    |        |
O   FORCES 18 |  ****| *   |    * |         |              |  *  *  |       |* *     |
N   HRNESS 13 |      |     |    * |         |              |        |       |***  ***|
    INTEST  6 |   *  |   * |      |         | *            |        |       |        |
B   JBARTZ 15 |  * **|     |    * |         |              |     *  | *     |  *     |
L   RSAVE  13 |      | * * |      |*        |              |        |       |  *   * |
O   ----------|------|-----|------|---------|--------------|--------|-------|--------|
C   SGMNTS 45 |  ****| *   | *    |** *  ***|*        *    |  *  *  |  *    |**    **|
K   TABLES 25 |      | *   |  * * |         |              |     * *|** ** *|*  * * *|
S   TEMPVI  7 |      |     |      |         |              |  *     |      *|        |
    TEMPVS 57 |  ****|*****| **   |*********|         *    |  *  *  |***    |***  ***|
    TITLES 17 |*  *  |   * |      |         |              |     *  | *     |  *  *  |
    VPOSTN  5 |      |     |      |         |              |        |       |        |
    ----------|------|-----|------|---------|--------------|--------|-------|--------|
    WINDFR  7 |      |     |    * |         |              |        | *     |        |
```

CALLING SUBROUTINES

CALLED COMMON BLOCKS

```
          :HHHHHHHHHHHHHHHH:IIII:KLLL:MMOO:PPPPPPPPPP:QQRRRR:SSSSSSSSSS:TUUU:VVVV:WXY:
          :YYYYYYYYYYYYYYYY:MMNN:IIOT:AARU:ADLLLLLORR:SUCOOS:EEEEILOOPP:RNPP:EIII:IDP:
          :ABBDEFLLLNPRSVVV:PPIT:NNGI:TTTT:NAEERSTSII:EARTTT:AGTTNPLLDL:IIDD:HNSS:NYR:
          :BNOASCIPPTEEOABF:LUTE:PAAM:33HP:EUDLEEXTNP:TTT AA:RSUUPLVVAI:GTAF:PPCP:D D:
          :FDXDTNMRXRNALLXN:SLAR:UXXE:130U:LXGPAGYPTL:    TR:CEPPUOARMN:F1TD:OUOR:Y E:
          :                :2SLS:TSS :   T:    FZR T:    ET:HG12TT  PE:S EC:STS :  G:
----------:----------------:----:----:----:----------:------:----------:----:----:---:
ABDATA   5:                :    :    :    :          :     *:          :    :    :   :
CDINT    7:                :    :    :    :       *  :     *:          :*   :    :   :
CEULER  14:                :  * :    :    : *      * :    **:  *       :  * :   *:   :
CMATRX  20:                :**  :    :   *:        * :     *:  **      :  * :   *:   :
CNSNTS  45:                :  **:*   :   *:       ** : * * *:   **     :    :** *:* *:
CNTSRF  19:                :    :    :    :   *     *:    **: *  *     : *  :    :*  :
----------:----------------:----:----:----:----------:------:----------:----:----:---:
COMAIN   6:                :    :    :   *:          :     *:          :    :    :   :
CONTRL  57:                :*** :*   :   *:**    ****:    **:  ***   * : ** :** *:*  :
CSTRNT  19:                :**  :    :    :   * *  * :    **: * **     :  * :    :   :
CYDATA   5:                :    :    :    :          :     *:          :    :    :   :
DAMPER   7:                :    :    :   *:       *  :    **:    *   * :    :    :   :
DESCRP  33:                :*** :    :   *: *      **:    **:  **      :  * : * *:   :
----------:----------------:----:----:----:----------:------:----------:----:----:---:
FLXBLE   9:                :**  :    :    : *        :     *:          :    :    :   :
FORCES  18:                :    :    :   *:   *   *  :     *: *      * :  * :   *:   :
HRNESS  13:                :    :    :   *:       * *:    **:          :  * :    :   :
INTEST   6:                :    :    :    : *        :     *:          :    : *  :   :
JBARTZ  15:                : *  :    :   *:       * *:    **:          : ** :    :   :
RSAVE   13:                :    :    :   *:   *   ** :    **: *        : *  :    :   :
----------:----------------:----:----:----:----------:------:----------:----:----:---:
SGMNTS  45:                :*** :    :   *:** * ** **:    **: ****   * : ** :** *:*  :
TABLES  25:                : *  :*   :    :   * *    :    **: *      * :  **:    :*  :
TEMPVI   7:                : *  :    :    :     *    :     *:          :  * :   *:   :
TEMPVS  57:    *   **      :* * :*   :   *: *** *****:    * : ****   * : *  : * *:*  :
TITLES  17:                :  * :*   :   *: *     ***:     *:    *     :    : *  :   :
VPOSTN   5:                :  * :    :    :      *   :     *:          :    :**  :   :
----------:----------------:----:----:----:----------:------:----------:----:----:---:
WINDFR   7:                :    :    :   *:          :    **:    *     :    :    :*  :
```

SUBROUTINE CROSS REFERENCE

CALLING SUBROUTINES

CALLED SUBROUTINES

```
           |MAAAAA|BBBBB|CCCCCC|DDDDDDDDD|DDDDDDDDDDDDDD|EEEEEEEE|FFFFFFG|HHHHHHHH|
           |ADIIII|EEGIL|FHIMOR|AAAAAAAAA|HIOOOORRRRSSSZ|DFJLLQUV|DILNRSL|BBEEIIPS|
           |IJRRRR|LLGNK|AANPNO|UUUUUUUUU|HNTTTTCCCIEEMP|EUOOTULA|INXTCMO|EPDRCNTE|
           |NUBBBB|TT PD|CIPUTS|XXXXXXXXX|PT33TTIQYFTTS |PNINIIRL|NPSEDSB|LLIRCPUT|
           |ASAGGG|GR UT|TNUTCS| 11233345|I 1333JUPTDQO |TCNGMLAF|IUERFOA|TANOSURC|
           | TG13G| T TA|T TET | 12212345|N   13KAR    L |HTT EBDD|TTGPLLL| YGNITB |
-----------|------|-----|------|---------|--------------|--------|-------|--------|
MAINA      |      |     |      |         |              |        |       |        |
ADJUST    1|      |     |      |         | *            |        |       |        |
AIRBAG    1|      |     |    * |         |              |        |       |        |
AIRBG1    1|      |     |      |         |              |        |       |        |
AIRBG3    1|      |     |      |         |              |        |       |        |
AIRBGG    2|  * * |     |      |         |              |        |       |        |
-----------|------|-----|------|---------|--------------|--------|-------|--------|
BELTG     1|      | *   |      |         |              |        |       |        |
BELTRT    1|      |     |    * |         |              |        |       |        |
BGG       1|     *|     |      |         |              |        |       |        |
BINPUT    1|*     |     |      |         |              |        |       |        |
BLKDTA    1|*     |     |      |         |              |        |       |        |
-----------|------|-----|------|---------|--------------|--------|-------|--------|
CFACTT    3|      |     |      |         |         ***  |        |       |        |
CHAIN     3|      |     |      |*        |              |     *  |       |        |
CINPUT    1|*     |     |      |         |              |        |       |        |
CMPUTE    1|      |     |      |         | *            |        |       |        |
CONTCT    1|      |     |      |*        |              |        |       |        |
CROSS    28|  *   |***  | *    |         |         *    |  *     |  *    |**      |
-----------|------|-----|------|---------|--------------|--------|-------|--------|
DAUX      5|      |     |      |         |              |     *  |       |        |
DAUX11    1|      |     |      |*        |              |        |       |        |
DAUX12    1|      |     |      |*        |              |        |       |        |
DAUX22    1|      |     |      |*        |              |        |       |        |
DAUX31    1|      |     |      |*        |              |        |       |        |
DAUX32    1|      |     |      |*        |              |        |       |        |
DAUX33    1|      |     |      |*        |              |        |       |        |
DAUX44    1|      |     |      |*        |              |        |       |        |
DAUX55    1|      |     |      |*        |              |        |       |        |
-----------|------|-----|------|---------|--------------|--------|-------|--------|
```

CALLING SUBROUTINES

CALLED SUBROUTINES

```
             |MAAAAA|BBBBB|CCCCCC|DDDDDDDDD|DDDDDDDDDDDDDD|EEEEEEEE|FFFFFFG|HHHHHHHH|
             |ADIIII|EEGIL|FHIMOR|AAAAAAAAA|HIOOOORRRRSSSZ|DFJLLQUV|DILNRSL|BBEEIIPS|
             |IJRRRR|LLGNK|AANPNO|UUUUUUUUU|HNTTTTCCCIEEMP|EUOOTULA|INXTCMO|EPDRCNTE|
             |NUBBBB|TT PD|CIPUTS|XXXXXXXXX|PT33TTIQYFTTS |PNINIIRL|NPSEDSB|LLIRCPUT|
             |ASAGGG|GR UT|TNUTCS| 11233345|I 1333JUPTDQO |TCNGMLAF|IUERFOA|TANOSURC|
             | TG13G| T TA|T TET | 12212345|N    13KAR   L|HTT EBDD|TTGPLLL| YGNITB |
-------------|------|-----|------|---------|--------------|--------|-------|--------|
DHHPIN      1|      |     |      |         |              |        |       |        |
DINT        1|*     |     |      |         |              |        |       |        |
DOT31      34|  ****| **  | *    |*  *     |         *    |  *  *  |  *    |**    **|
DOT33      10|   *  |     |      |         |      *       |  *     |  *    |        |
DOTT31      1|      |     |      |         |              |        |       |        |
DOTT33      9|      |     |      |         |              |        |  *    |        |
DRCIJK      2|      |     |      |         |              |     *  |       |        |
DRCQUA      1|      |     |      |         |              |        |       |        |
DRCYPR      6|   *  |   * |      |         |      *       |        |  *    |        |
DRIFT       1|      |     | *    |         |              |        |       |        |
DSETD       1|      |     |      |         |              |        |       |        |
DSETQ       1|      |     |      |         |              |        |       |        |
DSMSOL      4|      |     |      |         |              |*       |       |        |
DZP         1|      |     |   *  |         |              |        |       |        |
-------------|------|-----|------|---------|--------------|--------|-------|--------|
EDEPTH      1|      |  *  |      |         |              |        |       |        |
EFUNCT      2|      |     |      |         |              |  *     |       |        |
EJOINT      5|      |     |      |*        |         *    |        |       |        |
ELONG       1|      |*    |      |         |              |        |       |        |
ELTIME     40|* * * | * * | *  * |*********| *           *|  *  *  |  *  * |*     * |
EQUILB      1|      |     |      |         |              |        |       |        |
EULRAD      1|      |     |      |         |              |  *     |       |        |
EVALFD      7|      |     |      |         |              |        |*   *  |   *    |
-------------|------|-----|------|---------|--------------|--------|-------|--------|
FDINIT      2|      |     |      |         |              |        | *     |     *  |
FINPUT      1|      |     |  *   |         |              |        |       |        |
FLXSEG      1|      |     |      |*        |              |        |       |        |
FNTERP      1|      |     |      |         |              |        |       |        |
FRCDFL      7|      | *   |      |         |              |     *  |      *|*      *|
FSMSOL      2|      |     |      |*        |              |        |       |      * |
GLOBAL      2|      |     |      |         |              |  *     |       |        |
-------------|------|-----|------|---------|--------------|--------|-------|--------|
```

CALLING SUBROUTINES

CALLED SUBROUTINES

```
            :MAAAAA:BBBBB:CCCCCC:DDDDDDDDD:DDDDDDDDDDDDDD:EEEEEEEE:FFFFFFG:HHHHHHHH:
            :ADIIII:EEGIL:FHIMOR:AAAAAAAAA:HIOOOORRRRSSSZ:DFJLLQUV:DILNRSL:BBEEIIPS:
            :IJRRRR:LLGNK:AANPNO:UUUUUUUUU:HNTTTTCCCIEEMP:EUOOTULA:INXTCMO:EPDRCNTE:
            :NUBBBB:TT PD:CIPUTS:XXXXXXXXX:PT33TTIQYFTTS :PNINIIRL:NPSEDSB:LLIRCPUT:
            :ASAGGG:GR UT:TNUTCS: 11233345:I 1333JUPTDQO :TCNGMLAF:IUERFOA:TANOSURC:
            : TG13G: T TA:T TET : 12212345:N   13KAR   L :HTT EBDD:TTGPLLL: YGNITB :
------------:------:-----:------:---------:--------------:--------:-------:--------:
HBELT      2:      :     :    * :         :              :        :       :      * :
HBPLAY     2:      :     :      :         :              :        :       :      * :
HEDING     2:      :     :      :         :              :        :       :        :
HERRON     1:      :     :      :         :              :        :      *:        :
HICCSI     1:      :     :      :         :              :        :       :        :
HINPUT     1:      :     :  *   :         :              :        :       :        :
HPTURB     1:      :     :      :         :              :        :       :        :
HSETC      1:      :     :      :         :              :        :       :      * :
------------:------:-----:------:---------:--------------:--------:-------:--------:
HYABF      2:      :     :      :         :              :        :       :        :
HYBND      1:      :     :      :         :              :        :       :        :
HYBOX      1:      :     :      :         :              :        :       :        :
HYDAD      1:      :     :      :         :              :        :       :        :
HYEST      1:      :     :      :         :              :        :       :        :
HYFCN      3:      :     :      :         :              :        :       :        :
HYLIM      1:      :     :      :         :              :        :       :        :
HYLPR      1:      :     :      :         :              :        :       :        :
HYLPX      1:      :     :      :         :              :        :       :        :
HYNTR      1:      :     :      :         :              :        :       :        :
HYPEN      1:      :     :      :         :              :        :       :        :
HYREA      1:      :     :      :         :              :        :       :        :
HYSOL      1:      :     :      :         :              :        :       :        :
HYVAL      1:      :     :      :         :              :        :       :        :
HYVBX      1:      :     :      :         :              :        :       :        :
HYVFN      1:      :     :      :         :              :        :       :        :
------------:------:-----:------:---------:--------------:--------:-------:--------:
IMPLS2     1:      :     :      :         :              :        :       :        :
IMPULS     1:      :     :      :         :              :        :       :        :
INITAL     1:*     :     :      :         :              :        :       :        :
INTERS     2:      :  *  :      :         :              :        :       :        :
------------:------:-----:------:---------:--------------:--------:-------:--------:
```

CALLING SUBROUTINES

```
               :MAAAAA:BBBBB:CCCCCC:DDDDDDDDD:DDDDDDDDDDDDDD:EEEEEEEE:FFFFFFG:HHHHHHHH:
               :ADIIII:EEGIL:FHIMOR:AAAAAAAAA:HIOOOORRRRSSSZ:DFJLLQUV:DILNRSL:BBEEIIPS:
               :IJRRRR:LLGNK:AANPNO:UUUUUUUUU:HNTTTTCCCIEEMP:EUOOTULA:INXTCMO:EPDRCNTE:
               :NUBBBB:TT PD:CIPUTS:XXXXXXXXX:PT33TTIQYFTTS :PNINIIRL:NPSEDSB:LLIRCPUT:
               :ASAGGG:GR UT:TNUTCS: 11233345:I 1333JUPTDQO :TCNGMLAF:IUERFOA:TANOSURC:
               : TG13G: T TA:T TET : 12212345:N    13KAR   L :HTT EBDD:TTGPLLL: YGNITB :
    -----------:------:-----:------:---------:--------------:--------:-------:--------:
    KINPUT    1:      :     :  *   :         :              :        :       :        :
    LINAXS    1:      :     :      :         :              :        :       :        :
    LOGAXS    1:      :     :      :         :              :        :       :        :
    LTIME     1:      :     :      :         :              :    *   :       :        :
    -----------:------:-----:------:---------:--------------:--------:-------:--------:
C   MAT31    32:  * * :***  :      :*        :         *    :  *  *  :  *    :**    **:
A   MAT33    12:   *  :  *  :      :         :      *       :*       :  *    :        :
L   ORTHO     1:      :  *  :      :         :              :        :       :        :
L   OUTPUT    6:      :     :   *  :         : *            :     *  :       :        :
E   -----------:------:-----:------:---------:--------------:--------:-------:--------:
D   PANEL     2:   ** :     :      :         :              :        :       :        :
    PDAUX     2:      :     :   *  :         : *            :        :       :        :
S   PL[illegible] 1:      :     :      :         :              :        :       :        :
U   PLELP     3:      :     :    * :         :              :     *  :       :        :
B   PLREA     1:      :     :      :         :              :        :       :        :
R   PLSEGF    2:      :     :      :         :              :        :       :        :
O   PLTXYZ    1:      :     :      :         :              :        :       :        :
U   POSTPR    1:*     :     :      :         :              :        :       :        :
T   PRINT     6:*     :     :      :*        :              :     *  :       :        :
I   PRIPLT    1:*     :     :      :         :              :        :       :        :
N   -----------:------:-----:------:---------:--------------:--------:-------:--------:
E   QSET      1:      :     :      :         : *            :        :       :        :
S   QUAT      1:      :     :      :         :              :        :       :        :
    RCRT      1:      :  *  :      :         :              :        :       :        :
    ROT       2:      :     :      :         :        *     :  *     :       :        :
    ROTATE    1:      :     :      :         :              :        :       :        :
    RSTART    1:*     :     :      :         :              :        :       :        :
    -----------:------:-----:------:---------:--------------:--------:-------:--------:
```

CALLING SUBROUTINES

CALLED SUBROUTINES

```
            :MAAAAA:BBBBB:CCCCCC:DDDDDDDDD:DDDDDDDDDDDDDD:EEEEEEEE:FFFFFFG:HHHHHHHH:
            :ADIIII:EEGIL:FHIMOR:AAAAAAAAA:HIOOOORRRRSSSZ:DFJLLQUV:DILNRSL:BBEEIIPS:
            :IJRRRR:LLGNK:AANPNO:UUUUUUUUU:HNTTTTCCCIEEMP:EUOOTULA:INXTCMO:EPDRCNTE:
            :NUBBBB:TT PD:CIPUTS:XXXXXXXXX:PT33TTIQYFTTS :PNINIIRL:NPSEDSB:LLIRCPUT:
            :ASAGGG:GR UT:TNUTCS: 11233345:I 1333JUPTDQO :TCNGMLAF:IUERFOA:TANOSURC:
            : TG13G: T TA:T TET : 12212345:N   13KAR    L:HTT EBDD:TTGPLLL: YGNITB :
------------:------:-----:------:---------:--------------:--------:-------:--------:
SEARCH     1:      :     :      :         :              :        :       :        :
SEGSEG     2:      :     :    * :         :              :        :       :        :
SETUP1     1:      :     :      :*        :              :        :       :        :
SETUP2     2:      :     :      :*        :              :        :       :        :
SINPUT     1:*     :     :      :         :              :        :       :        :
SLPLOT     1:      :     :      :         :              :        :       :        :
SOLVA      1:      :     :      :         :              :        :       :        :
SOLVR      1:      :     :      :         :              :        :       :        :
SPDAMP     1:      :     :    * :         :              :        :       :        :
SPLINE     1:      :     :      :         :              :        :       :        :
------------:------:-----:------:---------:--------------:--------:-------:--------:
TRIGFS     1:      :     :      :         : *            :        :       :        :
UNIT1      2:*     :     :      :         :              :        :       :        :
UPDATE     1:      :     :      :         : *            :        :       :        :
UPDFDC     1:      :     :      :         :              :        :       :        :
------------:------:-----:------:---------:--------------:--------:-------:--------:
VEHPOS     2:      :     :      :*        :              :        :       :        :
VINPUT     1:*     :     :      :         :              :        :       :        :
VISCOS     2:      :     :      :         :              :  *     :       :        :
VISPR      2:      :     :      :*        :              :        :       :        :
------------:------:-----:------:---------:--------------:--------:-------:--------:
WINDY      1:      :     :    * :         :              :        :       :        :
XDY        8:      :     :      :         :              :     *  :  *    : *   ** :
YPRDEG     5:    * :     :      :         :              :        :       :        :
------------:------:-----:------:---------:--------------:--------:-------:--------:
```

CALLING SUBROUTINES

CALLED SUBROUTINES

```
           :HHHHHHHHHHHHHHHH:IIII:KLLL:MMOO:PPPPPPPPPP:QQRRRR:SSSSSSSSSS:TUUU:VVVV:WXY:
           :YYYYYYYYYYYYYYYY:MMNN:IIOT:AARU:ADLLLLLORR:SUCOOS:EEEEILOOPP:RNPP:EIII:IDP:
           :ABBDEFLLLNPRSVVV:PPIT:NNGI:TTTT:NAEERSTSII:EARTTT:AGTTNPLLDL:IIDD:HNSS:NYR:
           :BNOASCIPPTEEOABF:LUTE:PAAM:33HP:EUDLEEXTNP:TTT AA:RSUUPLVVAI:GTAF:PPCP:D D:
           :FDXDTNMRXRNALLXN:SLAR:UXXE:130U:LXGPAGYPTL:    TR:CEPPUOARMN:F1TD:OUOR:Y E:
           :                :2SLS:TSS :   T:     FZR T:    ET:HG12TT  PE:S EC:STS :  G:
-----------:----------------:----:----:----:----------:------:----------:----:----:---:
MAINA      :                :    :    :    :          :      :          :    :    :   :
ADJUST    1:                :    :    :    :          :      :          :    :    :   :
AIRBAG    1:                :    :    :    :          :      :          :    :    :   :
AIRBG1    1:                :    :    :    :          :      :    *     :    :    :   :
AIRBG3    1:                :    :    :    :          :      :          :  * :    :   :
AIRBGG    2:                :    :    :    :          :      :          :    :    :   :
-----------:----------------:----:----:----:----------:------:----------:----:----:---:
BELTG     1:                :    :    :    :          :      :          :    :    :   :
BELTRT    1:                :    :    :    :          :      :          :    :    :   :
BGG       1:                :    :    :    :          :      :          :    :    :   :
BINPUT    1:                :    :    :    :          :      :          :    :    :   :
BLKDTA    1:                :    :    :    :          :      :          :    :    :   :
-----------:----------------:----:----:----:----------:------:----------:----:----:---:
CFACTT    3:                :    :    :    :          :      :          :    :    :   :
CHAIN     3:                :  * :    :    :          :      :          :    :    :   :
CINPUT    1:                :    :    :    :          :      :          :    :    :   :
CMPUTE    1:                :    :    :    :          :      :          :    :    :   :
CONTCT    1:                :    :    :    :          :      :          :    :    :   :
CROSS    28:      *         : ** :    :   *:** * *    : *    : ****   * :  * : * *:*  :
-----------:----------------:----:----:----:----------:------:----------:----:----:---:
DAUX      5:                :**  :    :    : *        :      :          :  * :    :   :
DAUX11    1:                :    :    :    :          :      :          :    :    :   :
DAUX12    1:                :    :    :    :          :      :          :    :    :   :
DAUX22    1:                :    :    :    :          :      :          :    :    :   :
DAUX31    1:                :    :    :    :          :      :          :    :    :   :
DAUX32    1:                :    :    :    :          :      :          :    :    :   :
DAUX33    1:                :    :    :    :          :      :          :    :    :   :
DAUX44    1:                :    :    :    :          :      :          :    :    :   :
DAUX55    1:                :    :    :    :          :      :          :    :    :   :
-----------:----------------:----:----:----:----------:------:----------:----:----:---:
```

CALLING SUBROUTINES

CALLED SUBROUTINES

```
          :HHHHHHHHHHHHHHHH:IIII:KLLL:MMOO:PPPPPPPPPP:QQRRRR:SSSSSSSSSS:TUUU:VVVV:WXY:
          :YYYYYYYYYYYYYYYY:MMNN:IIOT:AARU:ADLLLLLORR:SUCOOS:EEEEILOOPP:RNPP:EIII:IDP:
          :ABBDEFLLLNPRSVVV:PPIT:NNGI:TTTT:NAEERSTSII:EARTTT:AGTTNPLLDL:IIDD:HNSS:NYR:
          :BNOASCIPPTEEOABF:LUTE:PAAM:33HP:EUDLEEXTNP:TTT AA:RSUUPLVVAI:GTAF:PPCP:D D:
          :FDXDTNMRXRNALLXN:SLAR:UXXE:13OU:LXGPAGYPTL:    TR:CEPPUOARMN:F1TD:OUOR:Y E:
          :                :2SLS:TSS :   T:     FZR T:    ET:HG12TT  PE:S EC:STS :  G:
----------:----------------:----:----:----:----------:------:----------:----:----:---:
DHHPIN   1:                :    :    :    :          :      :   *      :    :    :   :
DINT     1:                :    :    :    :          :      :          :    :    :   :
DOT31   34:         *      :*** :    :   *:*  * *  **:      : ***    * :  * :    :*  :
DOT33   10:                :*   :    :   *:        * :      :          : *  :   *:*  :
DOTT31   1:                :    :    :    :          :      :   *      :    :    :   :
DOTT33   9:    *    *      :    :    :   *:   *      :    * : * *      :    :    :*  :
DRCIJK   2:                :  * :    :    :          :      :          :    :    :   :
DRCQUA   1:                :    :    :    :          :      :          :    : *  :   :
DRCYPR   6:                :    :    :    :          :      :    *     :    : *  :   :
DRIFT    1:                :    :    :    :          :      :          :    :    :   :
DSETD    1:                :    :    :    :          :      :          :    : *  :   :
DSETQ    1:                :    :    :    : *        :      :          :    :    :   :
DSMSOL   4:                :*  *:    :    :          :      : *        :    :    :   :
DZP      1:                :    :    :    :          :      :          :    :    :   :
----------:----------------:----:----:----:----------:------:----------:----:----:---:
EDEPTH   1:                :    :    :    :          :      :          :    :    :   :
EFUNCT   2:                :    :    :    :          :      :          :    :   *:   :
EJOINT   5:                : ** :    :    :          :      :          :  * :    :   :
ELONG    1:                :    :    :    :          :      :          :    :    :   :
ELTIME  40:                :*** :    :   *: * *   * *:     *: ***    * :  * :   *:*  :
EQUILB   1:                :  * :    :    :          :      :          :    :    :   :
EULRAD   1:                :    :    :    :          :      :          :    :    :   :
EVALFD   7:                :    :    :    :     *    :      :        * :   *:    :*  :
----------:----------------:----:----:----:----------:------:----------:----:----:---:
FDINIT   2:                :    :    :    :          :      :          :    :    :   :
FINPUT   1:                :    :    :    :          :      :          :    :    :   :
FLXSEG   1:                :    :    :    :          :      :          :    :    :   :
FNTERP   1:                :    :    :    :          :      :          :    :   *:   :
FRCDFL   7:                :    :    :    :     *    :      :          :   *:    :   :
FSMSOL   2:                :    :    :    :          :      :          :    :    :   :
GLOBAL   2:                :    :    :    :          :      :          :    :   *:   :
----------:----------------:----:----:----:----------:------:----------:----:----:---:
```

CALLING SUBROUTINES

CALLED SUBROUTINES

```
           :HHHHHHHHHHHHHHHH:IIII:KLLL:MMOO:PPPPPPPPPP:QQRRRR:SSSSSSSSSS:TUUU:VVVV:WXY:
           :YYYYYYYYYYYYYYYY:MMNN:IIOT:AARU:ADLLLLLORR:SUCOOS:EEEEILOOPP:RNPP:EIII:IDP:
           :ABBDEFLLLNPRSVVV:PPIT:NNGI:TTTT:NAEERSTSII:EARTTT:AGTTNPLLDL:IIDD:HNSS:NYR:
           :BNOASCIPPTEEOABF:LUTE:PAAM:33HP:EUDLEEXTNP:TTT AA:RSUUPLVVAI:GTAF:PPCP:D D:
           :FDXDTNMRXRNALLXN:SLAR:UXXE:13OU:LXGPAGYPTL:    TR:CEPPUOARMN:F1TD:OUOR:Y E:
           :                :2SLS:TSS :   T:     FZR T:    ET:HG12TT  PE:S EC:STS :  G:
-----------:----------------:----:----:----:----------:------:----------:----:----:---:
HBELT     2:                :    :    :    :          :      :          :    :    :   :
HBPLAY    2:                :  * :    :    :          :      :          :    :    :   :
HEDING    2:                :    :    :   *:       *  :      :          :    :    :   :
HERRON    1:                :    :    :    :          :      :          :    :    :   :
HICCSI    1:                :    :    :    :       *  :      :          :    :    :   :
HINPUT    1:                :    :    :    :          :      :          :    :    :   :
HPTURB    1:                :    :    :    :          :      :          :  * :    :   :
HSETC     1:                :    :    :    :          :      :          :    :    :   :
-----------:----------------:----:----:----:----------:------:----------:----:----:---:
HYABF     2:    *    *      :    :    :    :          :      :          :    :    :   :
HYBND     1:                :    :    :    :  *       :      :          :    :    :   :
HYBOX     1:                :    :    :    :  *       :      :          :    :    :   :
HYDAD     1:         *      :    :    :    :          :      :          :    :    :   :
HYEST     1:                :    :    :    :          :      : *        :    :    :   :
HYFCN     3:*     *        *:    :    :    :          :      :          :    :    :   :
HYLIM     1:                :    :    :    :  *       :      :          :    :    :   :
HYLPR     1:        *       :    :    :    :          :      :          :    :    :   :
HYLPX     1:    *           :    :    :    :          :      :          :    :    :   :
HYNTR     1:                :    :    :    :          :      : *        :    :    :   :
HYPEN     1:                :    :    :    :   *      :      :          :    :    :   :
HYREA     1:                :    :    :    :  *       :      :          :    :    :   :
HYSOL     1:         *      :    :    :    :          :      :          :    :    :   :
HYVAL     1:                :    :    :    :  *       :      :          :    :    :   :
HYVBX     1:             *  :    :    :    :          :      :          :    :    :   :
HYVFN     1:             *  :    :    :    :          :      :          :    :    :   :
-----------:----------------:----:----:----:----------:------:----------:----:----:---:
IMPLS2    1:                :    :    :    :          :      :          :  * :    :   :
IMPULS    1:                :    :    :    :          :      :          :  * :    :   :
INITAL    1:                :    :    :    :          :      :          :    :    :   :
INTERS    2:                :    :    :    :          :      : *        :    :    :   :
-----------:----------------:----:----:----:----------:------:----------:----:----:---:
```

CALLING SUBROUTINES

```
                  |HHHHHHHHHHHHHHHH|IIII|KLLL|MMOO|PPPPPPPPPP|QQRRRR|SSSSSSSSSS|TUUU|VVVV|WXY|
                  |YYYYYYYYYYYYYYYY|MMNN|IIOT|AARU|ADLLLLLORR|SUCOOS|EEEEILOOPP|RNPP|EIII|IDP|
                  |ABBDEFLLLNPRSVVV|PPIT|NNGI|TTTT|NAEERSTSII|EARTTT|AGTTNPLLDL|IIDD|HNSS|NYR|
                  |BNOASCIPPTEEOABF|LUTE|PAAM|33HP|EUDLEEXTNP|TTT AA|RSUUPLVVAI|GTAF|PPCP|D D|
                  |FDXDTNMRXRNALLXN|SLAR|UXXE|13OU|LXGPAGYPTL|    TR|CEPPUOARMN|F1TD|OUOR|Y E|
                  |                |2SLS|TSS |   T|     FZR T|    ET|HG12TT  PE|S EC|STS |  G|
      ------------|----------------|----|----|----|----------|------|----------|----|----|---|
      KINPUT     1|                |    |    |    |          |      |          |    |    |   |
      LINAXS     1|                |    |    |    |          |      |     *    |    |    |   |
      LOGAXS     1|                |    |    |    |          |      |     *    |    |    |   |
      LTIME      1|                |    |    |    |          |      |          |    |    |   |
      ------------|----------------|----|----|----|----------|------|----------|----|----|---|
C     MAT31     32|    * * **      | ***|    |   *|* ** **   |    * | *      * |    |   *|*  |
A     MAT33     12|    *           |    |    |    |* *       |    * | * *      |    |    |*  |
L     ORTHO      1|                |    |    |    |          |      |          |    |    |   |
L     OUTPUT     6|                | *  |    |    |          |     *|          |  * |    |   |
E     ------------|----------------|----|----|----|----------|------|----------|----|----|---|
D     PANEL      2|                |    |    |    |          |      |          |    |    |   |
      PDAUX      2|                |    |    |    |          |      |          |    |    |   |
S     PLEDG      1|                |    |    |    |   *      |      |          |    |    |   |
U     PLELP      3|                | *  |    |    |          |      |          |    |    |   |
B     PLREA      1|                |    |    |    |  *       |      |          |    |    |   |
R     PLSEGF     2|                |    |    |    |   *      |      | *        |    |    |   |
O     PLTXYZ     1|                |    |    |    |         *|      |          |    |    |   |
U     POSTPR     1|                |    |    |    |          |      |          |    |    |   |
T     PRINT      6|                |**  |    |    |          |      |          |  * |    |   |
I     PRIPLT     1|                |    |    |    |          |      |          |    |    |   |
N     ------------|----------------|----|----|----|----------|------|----------|----|----|---|
E     QSET       1|                |    |    |    |          |      |          |    |    |   |
S     QUAT       1|                |    |    |    |          |      |          |    | *  |   |
      RCRT       1|                |    |    |    |          |      |          |    |    |   |
      ROT        2|                |    |    |    |          |      |          |    |    |   |
      ROTATE     1|                |  * |    |    |          |      |          |    |    |   |
      RSTART     1|                |    |    |    |          |      |          |    |    |   |
      ------------|----------------|----|----|----|----------|------|----------|----|----|---|
```

CALLING SUBROUTINES

```
                            :HHHHHHHHHHHHHHHH:IIII:KLLL:MMOO:PPPPPPPPPP:QQRRRR:SSSSSSSSSS:TUUU:VVVV:WXY:
                            :YYYYYYYYYYYYYYYY:MMNN:IIOT:AARU:ADLLLLLORR:SUCOOS:EEEEILOOPP:RNPP:EIII:IDP:
                            :ABBDEFLLLNPRSVVV:PPIT:NNGI:TTTT:NAEERSTSII:EARTTT:AGTTNPLLDL:IIDD:HNSS:NYR:
                            :BNOASCIPPTEEOABF:LUTE:PAAM:33HP:EUDLEEXTNP:TTT AA:RSUUPLVVAI:GTAF:PPCP:D D:
                            :FDXDTNMRXRNALLXN:SLAR:UXXE:130U:LXGPAGYPTL:    TR:CEPPUOARMN:F1TD:OUOR:Y E:
                            :                :2SLS:TSS :   T:     FZR T:    ET:HG12TT  PE:S EC:STS :  G:
               -------------:----------------:----:----:----:----------:------:----------:----:----:---:
               SEARCH      1:                :    :    :    :          :     *:          :    :    :   :
               SEGSEG      2:                : *  :    :    :          :      :          :    :    :   :
               SETUP1      1:                :    :    :    :          :      :          :    :    :   :
               SETUP2      2:                :    :    :    :          :      :          :  * :    :   :
               SINPUT      1:                :    :    :    :          :      :          :    :    :   :
         C     SLPLOT      1:                :    :    :    :       *  :      :          :    :    :   :
         A     SOLVA       1:                :    :    :    :          :      :          :    :    :*  :
         L     SOLVR       1:                :    :    :    :          :      :          :    :    :*  :
         L     SPDAMP      1:                :    :    :    :          :      :          :    :    :   :
         E     SPLINE      1:                :    :    :    :          :      :          :    : *  :   :
         D     -------------:----------------:----:----:----:----------:------:----------:----:----:---:
               TRIGFS      1:                :    :    :    :          :      :          :    :    :   :
         S     UNIT1       2:                :  * :    :    :          :      :          :    :    :   :
         U     UPDATE      1:                :    :    :    :          :      :          :    :    :   :
         B     UPDFDC      1:                :    :    :    :          :      :          :  * :    :   :
         R     -------------:----------------:----:----:----:----------:------:----------:----:----:---:
         O     VEHPOS      2:                :  * :    :    :          :      :          :    :    :   :
         U     VINPUT      1:                :    :    :    :          :      :          :    :    :   :
         T     VISCOS      2:                :    :    :    :          :      :          :    :   *:   :
         I     VISPR       2:                : *  :    :    :          :      :          :    :    :   :
         N     -------------:----------------:----:----:----:----------:------:----------:----:----:---:
         E     WINDY       1:                :    :    :    :          :      :          :    :    :   :
         S     XDY         8:                :*   :    :    :          :      : *        :  * :    :   :
               YPRDEG      5:                :  * :    :   *:        * :      :          :    : *  :   :
               -------------:----------------:----:----:----:----------:------:----------:----:----:---:
```

4.0 LIST OF 130 SUBROUTINES THAT COMPRISE THE ATB-IV MODEL COMPUTER PROGRAM

The first subroutine is a list of the common blocks used by the program, the second is the main program followed by all of the remaining subroutines in alphabetical order. Each subroutine name is appended with its revision number followed by the date of the latest change to the subroutine. This same data and revision number appears on the second line of each subroutine in Section 5.

SUBPROGRAM & REV. NO.		DATE	SUBPROGRAM & REV. NO.		DATE
BDATA	IV	07/23/86	MAIN	IV	07/23/86
ADJUST	IV	07/23/86	AIRBAG	IV	07/24/86
AIRBGG	III.5	10/17/85	AIRBG1	IV	07/24/86
AIRBG3	IV	07/23/86	BELTG	IV	07/23/86
BELTRT	IV	07/23/86	BGG	IV	07/23/86
BINPUT	IV	07/24/86	BLKDTA	IV	07/23/86
CFACTT	3	05/31/73	CHAIN	IV	07/24/86
CINPUT	III.2	08/08/84	CMPUTE	III.2	08/08/84
CONTCT	III.2	08/08/84	CROSS	3	05/31/73
DAUX	IV	07/24/86	DAUX11	IV	07/24/86
DAUX12	IV	07/24/86	DAUX22	IV	07/24/86
DAUX31	IV	07/24/86	DAUX32	IV	07/24/86
DAUX33	IV	07/24/86	DAUX44	IV	07/24/86
DAUX55	IV	07/24/86	DHHPIN	IV	07/24/86
DINT	IV	07/23/86	DOTT31	17	12/20/76
DOTT33	17	01/03/77	DOT31	17	01/03/77
DOT33	17	01/03/77	DRCIJK	18	02/24/78
DRCQUA	III.5	07/31/85	DRCYPR	IV	07/23/86
DRIFT	IV	07/24/86	DSETD	IV	07/23/86
DSETQ	IV	07/23/86	DSMSOL	3	07/08/74
DZP	IV	07/23/86	EDEPTH	IV	07/23/86
EFUNCT	20	04/29/80	EJOINT	IV	07/24/86
ELONG	1	10/05/72	ELTIME	III.2	08/08/84
EQUILB	IV	02/01/88	EULRAD	IV	07/23/86
EVALFD	IV	07/23/86	FDINIT	III.2	08/08/84
FINPUT	IV	02/01/88	FLXSEG	IV	07/23/86
FNTERP	IV	04/10/87	FRCDFL	III.2	08/08/84
FSMSOL	III.2	08/08/84	GLOBAL	IV	07/24/86
HBELT	IV	02/01/88	HBPLAY	III.5	10/17/85
HEDING	IV	02/01/88	HERRON	IV	07/23/86
HICCSI	IV	10/08/87	HINPUT	IV	07/23/86
HPTURB	IV	07/23/86	HSETC	III.2	08/08/84
HYABF	IV	02/07/87	HYBND	IV	02/07/87
HYBOX	IV	02/07/87	HYDAD	IV	02/07/87
HYEST	IV	02/07/87	HYFCN	IV	02/07/87
HYLIM	IV	12/11/87	HYLPR	IV	02/07/87
HYLPX	IV	02/07/87	HYNTR	IV	02/07/87
HYPEN	IV	02/07/87	HYREA	IV	12/11/87
HYSOL	IV	02/01/88	HYVAL	IV	12/11/87
HYVBX	IV	02/07/87	HYVFN	IV	12/11/87
IMPLS2	IV	07/24/86	IMPULS	IV	07/24/86
INITIAL	IV	07/24/86	INTERS	IV	02/23/86
KINPUT	IV	07/23/86	LINAXS	18	02/28/78

SUBPROGRAM & REV NO.		DATE	SUBPROGRAM & REV NO.		DATE
LOGAXS	19	09/18/79	LTIME	III.2	08/08/84
MAT31	17	01/03/77	MAT33	17	01/03/77
ORTHO	3	05/31/73	OUTPUT	IV	02/01/88
PANEL	III.2	08/08/84	PDAUX	IV	07/24/86
PLEDG	IV	02/07/87	PLELP	IV	02/07/87
PLREA	IV	12/11/87	PLSEGF	III.5	09/03/85
PLTXYZ	III.5	05/30/85	POSTPR	IV	02/01/88
PRINT	IV	07/24/86	PRIPLT	IV	07/24/86
QSET	III.3	10/01/84	QUAT	IV	07/23/86
RCRT	3	07/19/73	ROT	IV	07/23/86
ROTATE	IV	02/20/87	RSTART	IV	07/24/86
SEARCH	IV	07/24/86	SEGSEG	IV	02/07/87
SETUP1	IV	07/24/86	SETUP2	IV	07/24/86
SINPUT	IV	02/20/87	SLPLOT	III.2	08/08/84
SOLVA	III.2	08/08/84	SOLVR	III.2	08/08/84
SPDAMP	IV	07/24/86	SPLINE	19	05/14/79
SPRNGF	IV	07/23/86	TRIGFS	19	08/05/78
UNIT1	IV	02/20/87	UPDATE	IV	07/24/86
UPDFDC	III.2	08/08/84	VEHPOS	IV	07/23/86
VINPUT	IV	07/24/86	VISCOS	19	10/23/78
VISPR	IV	02/01/88	WINDY	IV	07/23/86
XDY	IV	07/23/86	YPRDEG	IV	11/26/86

5.0 FORTRAN SOURCE CODE OF THE ATB-IV.0 PROGRAM

Each of the 130 ATB-IV subroutines are listed in this section. The second line of each subroutine contains the subroutine revision number and the date of the latest change to the subroutine. Columns 72-80 of each line contain the subroutine name unless the line is a new or changed line from the listing in Ref. 4. In these cases the name of the latest change is in columns 72-80. The first subroutine is the BLOCKDATA containing the COMMON blocks used by the program. The second routine is the MAIN program which controls the flow of the program. The remaining subroutines are listed alphabetically.

```
      BLOCK DATA                                                        DECKA
C                                                    REV IV    07/23/86TWOPI
      IMPLICIT REAL*8 (A-H,O-Z)                                         DECKA
      COMMON/CONTRL/ TIME,NSEG,NJNT,NPL,NBLT,NBAG,NVEH,NGRND,           DECKA
     *               NS,NQ,NSD,NFLX,NHRNSS,NWINDF,NJNTF,NPRT(36),NPG    PAGE
      COMMON/CNSNTS/ PI,RADIAN,G,THIRD,EPS(24),                         DECKA
     *               UNITL,UNITM,UNITT,GRAVTY(3),TWOPI                  TWOPI
      COMMON/JBARTZ/ MNPL(    30),MNBLT(    8),MNSEG(    30),MNBAG(    6), DECKA
     *               MPL(3,5,30),MBLT(3,5,8),MSEG(3,5,30),MBAG(3,10,6), DECKA
     *               NTPL( 5,30),NTBLT( 5,8),NTSEG( 5,30)               DECKA
      COMMON/TITLES/ DATE(3),COMENT(40),VPSTTL(20),BDYTTL(5),           DECKA
     *               BLTTTL(5,8),PLTTL(5,30),BAGTTL(5,6),SEG(30),       DECKA
     *               JOINT(30),CGS(30),JS(30)                           DECKA
      REAL DATE,COMENT,VPSTTL,BDYTTL,BLTTTL,PLTTL,BAGTTL,SEG,JOINT      DECKA
      LOGICAL*1 CGS,JS                                                  DECKA
      COMMON/FORCES/PSF(7,70),BSF(4,20),SSF(10,40),BAGSF(3,20),         NCFORC
     *               PRJNT(7,30),NPANEL(5),NPSF,NBSF,NSSF,NBGSF         DECKA
      COMMON/RSAVE/  XSG(3,20,3),DPMI(3,3,30),LPMI(30),                 ATBIII
     *               NSG(9),MSG(20,9),MCG,MCGIN(24,5),KREF(20,9)        TTHKREF
      COMMON/CDINT/ UU(4),GH(3,4),                                      DECKA
     *               E(3,240),FF(5,240),GG(5,240),Y(5,240),U(5,240),    DECKA
     *               H,HPRINT,TSAVE,TPRINT,TSTART,ICNT,IDBL,IFLAG       DECKA
      COMMON/DAMPER/ APSDM(3,20),APSDN(3,20),ASD(5,20),MSDM(20),MSDN(20)DECKA
      COMMON/HRNESS/ BAR(15,100),BB(100),BBDOT(100),PLOSS(2,100),       DECKA
     *               XLONG(20),HTIME(2),IBAR(5,100),NL(2,100),          DECKA
     *               NPTSPB(20),NPTPLY(20),NTHRNS(20),NBLTPH(5)         DECKA
C     NOTE:  FF REPLACES F.                                             DECKA
      LOGICAL*1 FREE                                                    SLIP
      COMMON/TEMPVS/ JTMPVS(24000),FREE(30)                             SLIP
      COMMON/SGMNTS/ D(3,3,30),WMEG(3,30),WMEGD(3,30),U1(3,30),U2(3,30),DECKA
     *               SEGLP(3,30),SEGLV(3,30),SEGLA(3,30),NSYM(30)       DECKA
      COMMON/DESCRP/ PHI(3,30),W(30),RW(30),SR(4,60),HA(3,60),HB(3,60), SLIP
     *               RPHI(3,30),HT(3,3,60),SPRING(5,90),VISC(7,90),     DECKA
     *               JNT(30),IPIN(30),ISING(30),IGLOB(30),JOINTF(30)    DECKA
      COMMON/CNTSRF/ PL(24,30),BELT(20,8),TPTS(6,8),BD(24,40)           EDGE
      COMMON/TABLES/MXNTI,MXNTB,MXTB1,MXTB2,NTI(50),NTAB(1250),TAB(4500)BUTLER2
      COMMON/VPOSTN/ ZPLT(3),SPLT(3),AXV(3,6),VATAB(6,501,6),           VEHICL
     *               VT0(6),VDT(6),TIMEV(6),OMEGV(6),NVTAB(6),INDXV(6)  DECKA
      COMMON/CMATRX/ V1(3,30),V2(3,30),V3(3,12),B12(3,3,60),A22(3,3,60),DECKA
     *               F(3,30),TQ(3,30),WJ(30),A11(3,3,30)                SLIP
      COMMON/CEULER/ IEULER(30),HIR(3,3,90),ANG(3,30),ANGD(3,30),       JDRIFT
     *               FE(3,30),TQE(3,30),CONST(5,30)                     JDRIFT
      COMMON/FLXBLE/ HF(4,12,8),B42(3,3,24),V4(3,8),NFLEX(3,8)          DECKA
      COMMON/CSTRNT/ A13(3,3,24),A23(3,3,24),B31(3,3,24),B32(3,3,24),   DECKA
     *               HHT(3,3,12),RK1(3,12),RK2(3,12),QQ(3,12),TQQ(3,12),DECKA
     *               RQQ(3,12),HQQ(3,12),SQQ(12),CFQQ(12),              DECKA
     *               KQ1(12),KQ2(12),KQTYPE(12)                         DECKA
      COMMON/TEMPVI/ CREST,TTI(3),R1I(3),R2I(3),JSTOP(4,2,30)           DECKA
      COMMON/INTEST/ SGTEST(3,4,30),XTEST(3,120),SEGT(120),REGT(120)    DECKA
      REAL     SEGT                                                     DECKA
```

```
      COMMON/COMAIN/ VAR(240),DER(240),DT,H0,HMAX,HMIN,RSTIME,           DECKA
     *               ISTEP,NSTEPS,NDINT,NEQ,IRSIN,IRSOUT                  DECKA
      COMMON/ABDATA/ ZDEP(3,5),DBR(3,3,5),DPVCTR(3,5),DEPLOY(3,5),       DECKA
     *               AB(3,5),B(9,4,5),ZR(3,4,5),BFB(3,4,5),DRR(9,4,5),   DECKA
     *               VBAGG(5),VSCS(5),SPRK(5),CK(5),CMASS(5),CYMIN(5),   DECKA
     *               CYMOUT(5),BAGPV(5),PD(5),VBAG(5),VOLBP(5),           DECKA
     *               PCYV(5),PCYMIN(5),PVBAG(5),TV1(3,4,5),TV2(3,10,5),  DECKA
     *               SWITCH(5),PYMOUT(5),SCALE(5),PREVT,IFULL(6)          DECKA
      COMMON/CYDATA/ CYTD(5),CYPA(5),CYSP(5),CYTO(5),CYVO(5),CYCD(5),    DECKA
     *               CYK(5),CYR(5),CYAT(5),CYPV(5),CYCDO(5),CYAO(5),      DECKA
     *               CYPO(5),CYSS(5),CYLO(5),CYC(5),CYRHOO(5),CYVMAX(5),DECKA
     *               CYORFC(5),CYRHO(5),CYT(5),CYP(5),CYV(5)              DECKA
      COMMON/WINDFR/ WTIME(30),QFU(3,5),QFV(3,5),WF(3,30),IWIND(30),     WINDOP
     *               MWSEG(7,30),NFVSEG(6),NFVNT(5),MOWSEG(30,30)         WINDOP
      END                                                                 DECKA
```

```
C                                                                       MAINA
C     AAMRL ARTICULATED TOTAL BODY (ATBIV) MODEL COMPUTER PROGRAM      ATBIV
C     DEVELOPED BY CALSPAN CORP. AND J&J TECHNOLOGIES INC.              BUTLER1
C                                                 REV IV    07/23/86TWOPI
C     MAIN PROGRAM                                                      MAINA
C                                                                       MAINA
C     PERFORMS CARD INPUT, PROGRAM INITIALIZATION,                      MAINA
C     CONTROL OF INTEGRATION LOOP AND OPTIONAL OUTPUT.                  MAINA
C                                                                       MAINA
      IMPLICIT REAL*8(A-H,O-Z)                                          MAINA
      COMMON/CONTRL/ TIME,NSEG,NJNT,NPL,NBLT,NBAG,NVEH,NGRND,           MAINA
     *               NS,NQ,NSD,NFLX,NHRNSS,NWINDF,NJNTF,NPRT(36),NPG    PAGE
      COMMON/TITLES/ DATE(3),COMENT(40),VPSTTL(20),BDYTTL(5),           MAINA
     *               BLTTTL(5,8),PLTTL(5,30),BAGTTL(5,6),SEG(30),       MAINA
     *               JOINT(30),CGS(30),JS(30)                           MAINA
      REAL DATE,COMENT,VPSTTL,BDYTTL,BLTTTL,PLTTL,BAGTTL,SEG,JOINT      MAINA
      LOGICAL*1 CGS,JS                                                  MAINA
      COMMON/CNSNTS/ PI,RADIAN,G,THIRD,EPS(24),                         MAINA
     *               UNITL,UNITM,UNITT,GRAVTY(3),TWOPI                  TWOPI
      COMMON/COMAIN/ VAR(240),DER(240),DT,H0,HMAX,HMIN,RSTIME,          MAINA
     *               ISTEP,NSTEPS,NDINT,NEQ,IRSIN,IRSOUT                MAINA
      LOGICAL NPRT1,NPRT2,NPRT3                                         MAINA
      CALL ELTIME(1, 1)                                                 MAINA
C                                                                       PECONV
C     MAKE THE OUTPUT FILES PRINTER CONTROL FILES FOR THE P&E           PECONV
C                                                                       PECONV
      CALL CARCON(6,1)                                                  PECONV
      CALL CARCON(2,1)                                                  PECONV
C                                                                       MAINA
C     WRITE PROLOGUE ON PRIMARY OUTPUT UNIT.                            MAINA
C                                                                       MAINA
      NPG=2                                                             PAGE
      WRITE(6,11)                                                       MAINA
   11 FORMAT(1H1,30X,'AAMRL ARTICULATED TOTAL BODY (ATB) MODEL',52X,    ATBIV
     *        'PAGE     1'////                                          PAGE
     * 31X,'DEVELOPED BY CALSPAN CORP., P.O. BOX 400, BUFFALO NY 14225'/BUTLER1
     * 31X,'AND BY J&J TECHNOLOGIES INC., ORCHARD PARK, NY 14127'    // EDGE
     * 31X,'FOR THE AIR FORCE ARMSTRONG AEROSPACE MEDICAL RESEARCH  ' / VEHICL
     * 31X,'LABORATORY, WRIGHT PATTERSON AIR FORCE BASE        '       /ATBIV
     * 31X,'UNDER CONTRACTS F33615-75C-5002,-78C-0516 AND -80C-05117' //BUTLER1
     * 31X,'AND FOR THE NATIONAL HIGHWAY TRAFFIC SAFETY ADMINISTRATION,'BUTLER1
     */31X,'U.S. DEPARTMENT OF TRANSPORTATION, UNDER CONTRACTS'   /     BUTLER1
     * 31X,'FH-11-7592, HS-053-2-485, HS-6-01300 AND HS-6-01410.' ////  BUTLER1
     * 31X,'PROGRAM DOCUMENTATION: NHTSA REPORT NOS. DOT-HS-801-507' /  BUTLER1
     * 31X,'THROUGH 510 (FORMERLY CALSPAN REPORT NO. ZQ-5180-L-1),' /   BUTLER1
     * 31X,'AVAILABLE FROM NTIS (ACCESSION NOS. PB-241692,3,4 AND 5),' /BUTLER1
     * 31X,'APPENDIXES A-J TO THE ABOVE (AVAILABLE FROM CALSPAN),' /    BUTLER1
     * 31X,'AND REPORT NOS. AMRL-TR-75-14 (NTIS NO. AD-A014 816),/      ATBIV
     * 31X,'AFAMRL-TR-80-14 (NTIS NO. AD-A088 029), AND'/               ATBIV
     * 31X,'AFAMRL-TR-83-073 (NTIS NO. AD-B079 184).'////
```

```
     * 31X,'PROGRAM ATB-IV, EXECUTED ON THE AAMRL/BB CONCURRENT'/        ATBIV
     * 31X,'3250 COMPUTER, WRIGHT-PATTERSON AFB, OHIO'////)              ATBIV
C                                                                       MAINA
C     INPUT CARDS A.1 AND A.2, TEST FOR RESTART.                        MAINA
C                                                                       MAINA
      CALL BLKDTA                                                       MAINA
      READ(5,12)  DATE,IRSIN,IRSOUT,RSTIME,COMENT                       MAINA
   12 FORMAT(3A4,2I4,F8.0/20A4/20A4)                                    MAINA
      WRITE(6,13)  DATE,IRSIN,IRSOUT,RSTIME,COMENT                      MAINA
   13 FORMAT(/////4X,3A4,'  IRSIN=',I4,'  IRSOUT=',I4,'  RSTIME =',F8.4,MAINA
     *       61X,'CARDS A'//1X,20A4/1X,20A4//)                          MAINA
      IF (IRSIN.NE.0) GO TO 18                                          MAINA
C                                                                       MAINA
C     INPUT CARDS A.3,A.4 AND A.5.                                      MAINA
C                                                                       MAINA
      READ(5,14)  UNITL,UNITM,UNITT,GRAVTY,G                            MAINA
   14 FORMAT(3A4,4F12.0)                                                MAINA
      IF  (G.EQ.0.0)  G = DSQRT(GRAVTY(1)**2+GRAVTY(2)**2+GRAVTY(3)**2) MAINA
      READ(5,15)  NDINT,NSTEPS,DT,H0,HMAX,HMIN,NPRT                     MAINA
   15 FORMAT(2I4,4F8.0/36I2)                                            MAINA
      WRITE(6,16)  UNITL,UNITM,UNITT,GRAVTY,G,                          CHGIII
     *              NDINT,NSTEPS,DT,H0,HMAX,HMIN                        MAINA
   16 FORMAT(5X,'UNITL = ',A4,5X,'UNITM = ',A4,5X,'UNITT = ',A4,        CHGIII
     *       5X,'GRAVITY VECTOR = (',F9.4,',',F9.4,',',F9.4,')',5X,'G ='CHGIII
     *F9.4,//,5X,'NDINT =',I4,5X,'NSTEPS =',I5,5X,'DT =',F8.6,          MAINA
     *       5X,'H0 =',F8.6,5X,'HMAX =',F8.6,5X,'HMIN =',F8.6)          MAINA
      WRITE(6,17)  (I,I=1,36),NPRT                                      MAINA
   17 FORMAT('0 NPRT ARRAY'/3X,36I3/3X,36I3)                            MAINA
      NPRT4 = NPRT(4)                                                   MAINA
      IF(NPRT(26).GT.6) STOP 93                                         TGMOD1
      IF  (NPRT(4).LT.0)  GO TO 50                                      MAINA
C                                                                       MAINA
C     CALL INPUT ROUTINES                                               MAINA
C                                                                       MAINA
      CALL BINPUT                                                       MAINA
      CALL VINPUT                                                       MAINA
      CALL SINPUT                                                       MAINA
      CALL CINPUT                                                       MAINA
C                                                                       MAINA
C     PROGRAM INITIALIZATION                                            MAINA
C                                                                       MAINA
      TIME = 0.0                                                        MAINA
      CALL INITAL                                                       MAINA
      GO TO 19                                                          MAINA
C                                                                       MAINA
C     READ INPUT DATA FROM RESTART TAPE AND WRITE NEW TAPE.             MAINA
C     THE FIVE FUNCTIONS OF SUBROUTINE RSTART ARE:                      MAINA
C        1. READ INPUT & INITIALIZATION RECORD FROM OLD RESTART TAPE.   MAINA
C        2. WRITE INPUT & INITIALIZATION RECORD ONTO NEW RESTART TAPE.  MAINA
C        3. READ TIME POINT RECORD FROM OLD RESTART TAPE.               MAINA
```

```
C          4. READ NEW INPUT DATA FROM INPUT STREAM FOR RESTART.          MAINA
C          5. WRITE TIME POINT RECORD ONTO NEW RESTART TAPE.              MAINA
C                                                                         MAINA
   18 CALL RSTART(1,IRSIN)                                                MAINA
      CALL RSTART(4,5)                                                    MAINA
      NPRT4 = NPRT(4)                                                     MAINA
   19 IF (IRSOUT.NE.0) CALL RSTART(2,IRSOUT)                              MAINA
C                                                                         MAINA
C     INTEGRATION LOOP - ADVANCE TIME BY EITHER INTEGRATING THROUGH       MAINA
C     SUBROUTINE DINT OR BY FETCHING TIME POINT RECORD FROM RESTART TAPEMAINA
C                                                                         MAINA
      TIME = 0.0                                                          MAINA
      ISTEP = 0                                                           MAINA
   20 IF (IRSIN.EQ.0) GO TO 23                                            MAINA
      IF (TIME.GT.RSTIME+0.5*DT) GO TO 23                                 MAINA
      IF (DABS(TIME-RSTIME).LT.0.5*DT) GO TO 21                           MAINA
      CALL RSTART(3,IRSIN)                                                MAINA
      GO TO 24                                                            MAINA
   21 CALL RSTART(4,5)                                                    MAINA
      IF  (NPRT(4).LT.0)  GO TO 50                                        MAINA
   23 CALL DINT                                                           MAINA
C                                                                         MAINA
C     OPTIONAL OUTPUT                                                     MAINA
C     1. PRINTER PLOT ON OUTPUT UNIT 2 CONTROLLED BY NPRT(5) & (6).       MAINA
C                                                                         MAINA
   24 CALL PRIPLT                                                         MAINA
C                                                                         MAINA
C     2. RESTART DATA ON UNIT IRSOUT CONTROLLED BY IRSOUT # 0.            MAINA
C                                                                         MAINA
      IF (IRSOUT.NE.0) CALL RSTART(5,IRSOUT)                              MAINA
C                                                                         MAINA
C     3. SUBROUTINE PRINT ON PRIMARY OUTPUT UNIT CONTROLLED BE NPRT(3).  MAINA
C                                                                         MAINA
      NPRT3 = (NPRT(3).EQ.1)                                              MAINA
      IF (NPRT(3).GT.1) NPRT3 = (MOD(ISTEP,NPRT(3)).EQ.0)                 MAINA
      IF (NPRT3) CALL PRINT(6HMAIN3D)                                     MAINA
C                                                                         MAINA
C     4. PROGRAM VIEW PLOT DATA ON UNIT 1 CONTROLLED BY NPRT(1).          MAINA
C                                                                         MAINA
      NPRT1 = (NPRT(1).EQ.1)                                              MAINA
      IF (NPRT(1).GT.1) NPRT1 = (MOD(ISTEP,NPRT(1)).EQ.0)                 MAINA
      IF  (NPRT1)  CALL UNIT1(0)                                          MAINA
C                                                                         MAINA
C     5. SUBROUTINE ELTIME ON PRIMARY OUTPUT UNIT CONTROLLED BY NPRT(2).MAINA
C                                                                         MAINA
      NPRT2 = (NPRT(2).EQ.1)                                              MAINA
      IF (NPRT(2).GT.1) NPRT2 = (MOD(ISTEP,NPRT(2)).EQ.0)                 MAINA
      IF (NPRT2) CALL ELTIME(NPG,1)                                       PAGE
C                                                                         MAINA
C     END OF INTEGRATION LOOP.                                            MAINA
```

```
C                                                                       MAINA
      ISTEP = ISTEP+1                                                   MAINA
      IF (ISTEP.LE.NSTEPS) GO TO 20                                     MAINA
C                                                                       MAINA
C     6. SUBROUTINE POSTPR ON PRIMARY OUTPUT UNIT CONTROLLED BY NPRT(4).MAINA
C                                                                       MAINA
   50 IF  (NPRT4.GT.0)  END FILE 8                                      MAINA
      IF  (NPRT(4).EQ.0 .OR. NPRT(4).EQ.4)  GO TO 60                    MAINA
      PRDT = 1000.0*DT                                                  MAINA
      CALL POSTPR (PRDT)                                                MAINA
      IF  (NPRT2)  CALL ELTIME(NPG,1)                                   PAGE
C                                                                       MAINA
C     7. END OF RUN - CALL ELTIME IF NOT CALLED ABOVE.                  MAINA
C                                                                       MAINA
   60 IF  (.NOT.NPRT2)  CALL ELTIME(NPG,1)                              PAGE
      STOP  1                                                           MAINA
      END                                                               MAINA
```

```
      SUBROUTINE ADJUST (M,D1)                                          ADJUST
C                                                    REV IV    07/23/86TWOPI
      IMPLICIT REAL*8 (A-H,O-Z)                                         ADJUST
      COMMON/CNSNTS/ PI,RADIAN,G,THIRD,EPS(24),                         ADJUST
     *               UNITL,UNITM,UNITT,GRAVTY(3),TWOPI                  TWOPI
      COMMON/CDINT/ UU(4),GH(3,4),                                      ADJUST
     *              E(3,240), F(5,240),GG(5,240),Y(5,240),U(5,240),     ADJUST
     *              H,HPRINT,HS,TPRINT,TSTART,ICNT,IDBL,IFLAG           ADJUST
      COMMON/COMAIN/ VAR(240),DER(240),DT,HO,EMAX,HMIN,RSTIME,          ADJUST
     *               ISTEP,NSTEPS,NDINT,NEQ,IRSIN,IRSOUT                ADJUST
      IF  (M.NE.1)  GO TO 12                                            ADJUST
C                                                                       ADJUST
C     M = 1:                                                            ADJUST
C                                                                       ADJUST
      DO  11  I=1,NEQ                                                   ADJUST
      W = VAR(I) - GG(1,I)                                              ADJUST
      Z = DER(I) - GG(2,I)                                              ADJUST
      ZZ = Z - GG(5,I)*W - GG(3,I)*UU(3) - GG(4,I)*UU(4)                ADJUST
      GG(3,I) = GG(3,I) + ZZ*UU(1)                                      ADJUST
      GG(4,I) = GG(4,I) + ZZ*UU(2)                                      ADJUST
      Y(1,I) = VAR(I)                                                   ADJUST
   11 Y(2,I) = DER(I)                                                   ADJUST
      GO TO 99                                                          ADJUST
   12 IF  (M.EQ.3)  GO TO 23                                            ADJUST
C                                                                       ADJUST
C     M = 2,4,5:                                                        ADJUST
C                                                                       ADJUST
      H1 = EPS(1)/H                                                     ADJUST
      N2 = NEQ/2                                                        ADJUST
      DO  20  I=1,NEQ,3                                                 ADJUST
      ZA = 0.0                                                          ADJUST
      IF  (I.LE.N2)  GO TO 20                                           ADJUST
      IF  (M.EQ.4)   GO TO 16                                           ADJUST
      VARX = VAR(I  ) - Y(1,I  )                                        ADJUST
      VARY = VAR(I+1) - Y(1,I+1)                                        ADJUST
      VARZ = VAR(I+2) - Y(1,I+2)                                        ADJUST
      DERX = DER(I  ) - Y(2,I  )                                        ADJUST
      DERY = DER(I+1) - Y(2,I+1)                                        ADJUST
      DERZ = DER(I+2) - Y(2,I+2)                                        ADJUST
      GO TO 17                                                          ADJUST
   16 VARX = VAR(I  ) - U(1,I  )                                        ADJUST
      VARY = VAR(I+1) - U(1,I+1)                                        ADJUST
      VARZ = VAR(I+2) - U(1,I+2)                                        ADJUST
      DERX = DER(I  ) - U(2,I  )                                        ADJUST
      DERY = DER(I+1) - U(2,I+1)                                        ADJUST
      DERZ = DER(I+2) - U(2,I+2)                                        ADJUST
   17 U(3,I) = U(3,I) + VARX*DERX + VARY*DERY + VARZ*DERZ               ADJUST
      U(4,I) = U(4,I) + VARX**2 + VARY**2 + VARZ**2                     ADJUST
      IF (U(4,I).EQ.0.0) GO TO 18                                       FIXADJ
      ZA = H1                                                           FIXADJ
```

```
      IF (U(3,I).LT.H1*U(4,I))  ZA = U(3,I)/U(4,I)                      FIXADJ
   18 GG(5,I+2) = ZA                                                    FIXADJ
      GG(5,I+1) = ZA                                                    ADJUST
   20 GG(5,I  ) = ZA                                                    ADJUST
      GO TO (99,21,99,23,25),M                                          ADJUST
C                                                                       ADJUST
C     M = 2:                                                            ADJUST
C                                                                       ADJUST
   21 DO  22  I=1,NEQ                                                   ADJUST
      ZA = GG(5,I)                                                      ADJUST
      Y1 = Y(4,I)   - ZA*Y(3,I)                                         ADJUST
      Y2 = GG(2,I)  - ZA*GG(1,I)                                        ADJUST
      Y3 = DER(I)   - ZA*VAR(I)                                         ADJUST
      GG(3,I) = -Y1*GH(1,1) + Y2*GH(2,1) + Y3*GH(3,1)                   ADJUST
      GG(4,I) =  Y1*GH(1,2) - Y2*GH(2,2) + Y3*GH(3,2)                   ADJUST
      Y(1,I) = 0.5*(Y(1,I)+VAR(I))                                      ADJUST
   22 Y(2,I) = 0.5*(Y(2,I)+DER(I))                                      ADJUST
      GO TO 99                                                          ADJUST
C                                                                       ADJUST
C     M = 3,4:                                                          ADJUST
C                                                                       ADJUST
   23 DO  24  I=1,NEQ                                                   ADJUST
      ZA = GG(5,I)                                                      ADJUST
      Y1 = GG(2,I) - ZA*GG(1,I)                                         ADJUST
      Y2 = Y(2,I)  - ZA*Y(1,I)                                          ADJUST
      Y3 = DER(I)  - ZA*VAR(I)                                          ADJUST
      GG(3,I) = -Y1*GH(1,3) + Y2*GH(2,3) - Y3*GH(3,3)                   ADJUST
      GG(4,I) =  Y1*GH(1,4) - Y2*GH(2,4) + Y3*GH(3,4)                   ADJUST
      U(1,I) = VAR(I)                                                   ADJUST
   24 U(2,I) = DER(I)                                                   ADJUST
      GO TO 99                                                          ADJUST
C                                                                       ADJUST
C     M = 5:                                                            ADJUST
C                                                                       ADJUST
   25 DO  26  I=1,NEQ                                                   ADJUST
      ZA = GG(5,I)                                                      ADJUST
      Y1 = GG(2,I) - ZA*GG(1,I)                                         ADJUST
      Y2 = DER(I)  - ZA*VAR(I)                                          ADJUST
      Y3 = U(2,I)  - ZA*U(1,I)                                          ADJUST
      GG(3,I) = -Y1*GH(1,3) + Y2*GH(2,3) - Y3*GH(3,3)                   ADJUST
      GG(4,I) =  Y1*GH(1,4) - Y2*GH(2,4) + Y3*GH(3,4)                   ADJUST
      Y(1,I) = VAR(I)                                                   ADJUST
   26 Y(2,I) = DER(I)                                                   ADJUST
   99 RETURN                                                            ADJUST
      END                                                               ADJUST
```

```
      SUBROUTINE  AIRBAG                                                AIRBAG
C                                                 REV IV    07/24/86SLIP
C     AIRBAG ROUTINE CALLED BY SUBROUTINE CONTCT TO DETERMINE THE INTER-AIRBAG
C     ACTION OF THE BAG WITH REACTION PANELS AND BODY SEGMENTS BY USE OFAIRBAG
C     SUBROUTINE BGG. THE DIFFERENTIAL PRESSURE,FORCE AND TORQUE ON THE AIRBAG
C     BAG IS EVALUATED AND THE RESULTING FORCE AND TORQUE ON THE BODY   AIRBAG
C     SEGMENTS ARE ADDED TO THE U1 AND U2 ARRAYS.                       AIRBAG
C                                                                       AIRBAG
      IMPLICIT REAL*8 (A-H,O-Z)                                         AIRBAG
      COMMON/CONTRL/ TIME,NSEG,NJNT,NPL,NBLT,NBAG,NVEH,NGRND,           AIRBAG
     *               NS,NQ,NSD,NFLX,NHRNSS,NWINDF,NJNTF,NPRT(36),NPG    PAGE
      COMMON/SGMNTS/ D(3,3,30),WMEG(3,30),WMEGD(3,30),U1(3,30),U2(3,30),AIRBAG
     *               SEGLP(3,30),SEGLV(3,30),SEGLA(3,30),NSYM(30)       AIRBAG
      COMMON/DESCRP/ PHI(3,30),W(30),RW(30),SR(4,60),HA(3,60),HB(3,60), SLIP
     *               RPHI(3,30),HT(3,3,60),SPRING(5,90),VISC(7,90),     AIRBAG
     *               JNT(30),IPIN(30),ISING(30),IGLOB(30),JOINTF(30)    AIRBAG
      COMMON/JBARTZ/ MNPL(    30),MNBLT(    8),MNSEG(    30),MNBAG(     6), AIRBAG
     *               MPL(3,5,30),MBLT(3,5,8),MSEG(3,5,30),MBAG(3,10,6), AIRBAG
     *               NTPL( 5,30),NTBLT( 5,8),NTSEG( 5,30)               AIRBAG
      COMMON/FORCES/PSF(7,70),BSF(4,20),SSF(10,40),BAGSF(3,20),         NCFORC
     *               PRJNT(7,30),NPANEL(5),NPSF,NBSF,NSSF,NBGSF         AIRBAG
      COMMON/CNTSRF/ PL(24,30),BELT(20,8),TPTS(6,8),BD(24,40)           EDGE
      COMMON/CNSNTS/ PI,RADIAN,G,THIRD,EPS(24),                         AIRBAG
     *               UNITL,UNITM,UNITT,GRAVTY(3),TWOPI                  TWOPI
      COMMON/ABDATA/ ZDEP(3,5),DBR(3,3,5),DPVCTR(3,5),DEPLOY(3,5),      AIRBAG
     *              AB(3,5),B(9,4,5),ZR(3,4,5),BFB(3,4,5),DRR(9,4,5),   AIRBAG
     *              VBAGG(5),VSCS(5),SPRK(5),CK(5),CMASS(5),CYMIN(5),   AIRBAG
     *              CYMOUT(5),BAGPV(5),PD(5),VBAG(5),VOLBP(5),          AIRBAG
     *              PCYV(5),PCYMIN(5),PVBAG(5),TV1(3,4,5),TV2(3,10,5),  AIRBAG
     *              SWITCH(5),PYMOUT(5),SCALE(5),PREVT,IFULL(6)         AIRBAG
      COMMON/CYDATA/ CYTD(5),CYPA(5),CYSP(5),CYT0(5),CYV0(5),CYCD(5),   AIRBAG
     *              CYK(5),CYR(5),CYAT(5),CYPV(5),CYCD0(5),CYA0(5),     AIRBAG
     *              CYP0(5),CYSS(5),CYL0(5),CYC(5),CYRHO0(5),CYVMAX(5),AIRBAG
     *              CYORFC(5),CYRHO(5),CYT(5),CYP(5),CYV(5)             AIRBAG
      COMMON/TEMPVS/ TMP(9),TMP1(3),TORQ(3),FORCE(3,5),TORA(3,5),       AIRBAG
     *          TQB(3,10),FRB(3,10),VOL(10),DELF(3),VOLP(4,5),FRA(4,5)AIRBAG
C     NOTE: THIS COMMON/TEMPVS/ IS SHARED BY AIRBAG AND AIRBGG.         AIRBAG
      CALL ELTIME(1,24)                                                 AIRBAG
      DELT = TIME-PREVT                                                 AIRBAG
      NBGSF = 0                                                         AIRBAG
      DO  70  J=1,NBAG                                                  AIRBAG
      IF  (MNBAG(J).EQ.0)  GO TO 70                                     AIRBAG
      IF  (IFULL(J).LE.0)  GO TO 69                                     AIRBAG
      CALL AIRBGG(J)                                                    AIRBAG
C                                                                       AIRBAG
C     COMPUTE  CMOUT: MASS FLOW OUT OF BAG                              AIRBAG
C              BAGPV: UNDISTORTED BAG VOLUME                            AIRBAG
C                                                                       AIRBAG
      IF  (PD(J).GT.CYPV(J))  CYMOUT(J) = PYMOUT(J)                     AIRBAG
     *                         + DELT*CYORFC(J)*DSQRT(PD(J))            AIRBAG
```

```
      BAGPV(J) = CYPA(J)*((CYMIN(J)-CYMOUT(J))*SWITCH(J))**CYK(J)      AIRBAG
C                                                                      AIRBAG
C     BAG IS FULLY INFLATED, COMPUTE DIFFERENTIAL PRESSURE             AIRBAG
C                                                                      AIRBAG
      PD(J) = BAGPV(J)/(VBAG(J)-VOLBP(J))**CYK(J) - CYPA(J)           AIRBAG
      JB = NVEH + J                                                    AIRBAG
      KP = NPANEL(J)                                                   AIRBAG
      KBAG = MNBAG(J)                                                  AIRBAG
C                                                                      AIRBAG
C     OPTIONAL DIAGNOSTIC OUTPUT                                       AIRBAG
C                                                                      AIRBAG
      IF  (NPRT(21).NE.0)  WRITE(6,41)                                 AIRBAG
     *  ((FRB(I,K),I=1,3),(TQB(I,K),I=1,3),K=1,KBAG),(FORCE(I,J),I=1,3),AIRBAG
     *  (TORA(I,J),I=1,3),TORQ,((FRA(I,K),I=1,3),VOLP(K,J),K=1,KP),    AIRBAG
     *  (VOL(K),K=1,KBAG),VOLBP(J),CYMOUT(J),BAGPV(J),PD(J)            AIRBAG
   41 FORMAT ('0AIRBAG CONTCT'/(1X,9G14.6))                            AIRBAG
      IF  (PD(J).LT.0.0)  PD(J) = 0.0                                  AIRBAG
      IF  (PD(J).EQ.0.0)  GO TO 46                                     AIRBAG
C                                                                      AIRBAG
C     SET UP BAGSF ARRAY FOR OUTPUT ROUTINE                            AIRBAG
C                                                                      AIRBAG
      KBGSF = NBGSF+5                                                  AIRBAG
      DO  42  K=1,KP                                                   AIRBAG
      KBGSF = KBGSF+1                                                  AIRBAG
      DO  42  I=1,3                                                    AIRBAG
   42 BAGSF(I,KBGSF) = PD(J)*FRA(I,K)                                  AIRBAG
      DO  45  I=1,KBAG                                                 AIRBAG
      KBGSF = KBGSF+1                                                  AIRBAG
      IF  (VOL(I).EQ.0.0)  GO TO 45                                    AIRBAG
      M = MBAG(2,I,J)                                                  AIRBAG
C                                                                      AIRBAG
C     FINAL COMPUTATIONS OF FORCE AND TORQUE ON AIRBAG                 AIRBAG
C                                                                      AIRBAG
      DO  44  K=1,3                                                    AIRBAG
      FRB(K,I) = PD(J)*FRB(K,I)                                        AIRBAG
      BAGSF(K,KBGSF) = FRB(K,I)                                        AIRBAG
      U1(K,M) = U1(K,M) - FRB(K,I)                                     AIRBAG
   44 U2(K,M) = U2(K,M) + PD(J)*TQB(K,I)                               AIRBAG
   45 CONTINUE                                                         AIRBAG
   46 DO  47  K=1,3                                                    AIRBAG
      FORCE(K,J) = PD(J)*FORCE(K,J)                                    AIRBAG
   47 TORA (K,J) = PD(J)*TORA (K,J)                                    AIRBAG
      IF  (VOLP(1,J).NE.0.0)  GO TO 55                                 AIRBAG
C                                                                      AIRBAG
C     AIRBAG IS NOT INTERSECTING PRIMARY REACTION PANEL.               AIRBAG
C     COMPUTE ARTIFICIAL FORCE AND TORQUE WITH A LINEAR SPRING FUNCTION AIRBAG
C     IN AN ATTEMPT TO TIE +X SEMIAXIS ENDPOINT OF AIRBAG TO DEPLOYMENT AIRBAG
C     POINT ON REACTION PANEL.                                         AIRBAG
C                                                                      AIRBAG
      DO  51  K=1,3                                                    AIRBAG
```

```
51  TMP(K) = BFB(K,1,J) + ZDEP(K,J)                                    AIRBAG
    CALL DOT31 (D(1,1,NVEH),TMP,TMP1)                                  AIRBAG
    DO  52  K=1,3                                                      AIRBAG
    DELF(K) = TMP1(K) + SEGLP(K,NVEH) - SEGLP(K,JB)                    AIRBAG
52  TMP(K) = BD(K+3,JB)                                                AIRBAG
    TMP(1) = TMP(1) + BD(1,JB)                                         AIRBAG
    CALL DOT31 (D(1,1,JB),TMP,TMP1)                                    AIRBAG
    DO  53  K=1,3                                                      AIRBAG
    DELF(K) = SPRK(J)*(DELF(K)-TMP1(K))                                AIRBAG
    BAGSF(K,NBGSF+5) = DELF(K)                                         AIRBAG
53  FORCE(K,J) = FORCE(K,J) + DELF(K)                                  AIRBAG
    CALL MAT31 (D(1,1,JB),DELF,TMP1)                                   AIRBAG
    CALL CROSS (TMP,TMP1,DELF)                                         AIRBAG
    DO  54  K=1,3                                                      AIRBAG
54  TORA(K,J) = TORA(K,J) + DELF(K)                                    AIRBAG
55  XDD = CYMIN(J) - CYMOUT(J) + W(JB)                                 AIRBAG
    FMASS = CMASS(J)*XDD/G                                             AIRBAG
    TMASS = CMASS(J)*(XDD+W(JB)*2.0/3.0)/G                             AIRBAG
    DO  56  I=1,3                                                      AIRBAG
56  TMP(I) = WMEG(I,JB)*PHI(I,JB)                                      AIRBAG
    CALL CROSS (WMEG(1,JB),TMP,TMP1)                                   AIRBAG
    DO  57  I=1,3                                                      AIRBAG
    SEGLA(I,JB) = FORCE(I,J)/FMASS + GRAVTY(I)                         AIRBAG
57  WMEGD(I,JB) = (TORA(I,J)/TMASS-TMP1(I))*RPHI(I,JB)                 AIRBAG
69  NBGSF = NBGSF + 5 + NPANEL(J) + MNBAG(J)                           AIRBAG
70  CONTINUE                                                           AIRBAG
    CALL ELTIME(2,24)                                                  AIRBAG
    RETURN                                                             AIRBAG
    END                                                                AIRBAG
```

```
      SUBROUTINE AIRBGG(J)                                              AIRBGG
C                                                  REV III.5 10/17/85EDGE
C     CALLED BY SUBROUTINES AIRBAG AND AIRBG3 TO COMPUTE VOLUMES OF     AIRBGG
C     INTERSECTION BETWEEN AIRBAGS AND PANELS AND SEGMENTS.             AIRBGG
C                                                                       AIRBGG
      IMPLICIT REAL*8 (A-H,O-Z)                                         AIRBGG
      COMMON/CONTRL/ TIME,NSEG,NJNT,NPL,NBLT,NBAG,NVEH,NGRND,           AIRBGG
     *               NS,NQ,NSD,NFLX,NHRNSS,NWINDF,NJNTF,NPRT(36),NPG    PAGE
      COMMON/SGMNTS/ D(3,3,30),WMEG(3,30),WMEGD(3,30),U1(3,30),U2(3,30),AIRBGG
     *               SEGLP(3,30),SEGLV(3,30),SEGLA(3,30),NSYM(30)       AIRBGG
      COMMON/JBARTZ/ MNPL(   30),MNBLT(    8),MNSEG(   30),MNBAG(    6), AIRBGG
     *               MPL(3,5,30),MBLT(3,5,8),MSEG(3,5,30),MBAG(3,10,6), AIRBGG
     *               NTPL( 5,30),NTBLT( 5,8),NTSEG( 5,30)               AIRBGG
      COMMON/FORCES/PSF(7,70),BSF(4,20),SSF(10,40),BAGSF(3,20),         NCFORC
     *               PRJNT(7,30),NPANEL(5),NPSF,NBSF,NSSF,NBGSF         AIRBGG
      COMMON/CNTSRF/ PL(24,30),BELT(20,8),TPTS(6,8),BD(24,40)           EDGE
      COMMON/ABDATA/ ZDEP(3,5),DBR(3,3,5),DPVCTR(3,5),DEPLOY(3,5),      AIRBGG
     *               AB(3,5),B(9,4,5),ZR(3,4,5),BFB(3,4,5),DRR(9,4,5),  AIRBGG
     *               VBAGG(5),VSCS(5),SPRK(5),CK(5),CMASS(5),CYMIN(5),  AIRBGG
     *               CYMOUT(5),BAGPV(5),PD(5),VBAG(5),VOLBP(5),         AIRBGG
     *               PCYV(5),PCYMIN(5),PVBAG(5),TV1(3,4,5),TV2(3,10,5), AIRBGG
     *               SWITCH(5),PYMOUT(5),SCALE(5),PREVT,IFULL(6)        AIRBGG
      COMMON/CYDATA/ CYTD(5),CYPA(5),CYSP(5),CYTO(5),CYVO(5),CYCD(5),   AIRBGG
     *               CYK(5),CYR(5),CYAT(5),CYPV(5),CYCDO(5),CYAO(5),    AIRBGG
     *               CYPO(5),CYSS(5),CYLO(5),CYC(5),CYRHOO(5),CYVMAX(5),AIRBGG
     *               CYORFC(5),CYRHO(5),CYT(5),CYP(5),CYV(5)            AIRBGG
      COMMON/TEMPVS/ TMP(9),TMP1(3),TORQ(3),FORCE(3,5),TORA(3,5),       AIRBGG
     *          TQB(3,10),FRB(3,10),VOL(10),DELF(3),VOLP(4,5),FRA(4,5)AIRBGG
C     NOTE: THIS COMMON/TEMPVS/ IS SHARED BY AIRBAG AND AIRBGG.         AIRBGG
      JB = NVEH + J                                                     AIRBGG
      VOLBP(J) = 0.0                                                    AIRBGG
C                                                                       AIRBGG
C     COMPUTE THERMODYNAMIC PROPERTIES OF AIRBAG                        AIRBGG
C         CYRHO  : DENSITY                                              AIRBGG
C         CYT    : TEMPERATURE                                          AIRBGG
C         CYP    : PRESSURE                                             AIRBGG
C         CYMIN  : MASS FLOW INTO BAG                                   AIRBGG
C         VBCALC : CALCULATED VOLUME                                    AIRBGG
C                                                                       AIRBGG
      Q  = 1.0                                                          AIRBGG
      Q1 = 1.0                                                          AIRBGG
      Q2 = 1.0                                                          AIRBGG
      IF  (TIME.LE.CYTD(J))  GO TO 13                                   AIRBGG
      Q  = 1.0 + CYC(J)*(TIME-CYTD(J))                                  AIRBGG
      CYK1 = 2.0/(CYK(J)-1.0)                                           AIRBGG
      Q1 = 1.0/Q**CYK1                                                  AIRBGG
      Q2 = 1.0/Q**(CYK(J)*CYK1)                                         AIRBGG
   13 CYRHO(J) = CYRHOO(J)*Q1                                           AIRBGG
      CYT(J)   = CYTO(J)/Q**2                                           AIRBGG
      CYP(J)   = CYPO(J)*Q2                                             AIRBGG
```

```
      CYMIN(J) = CYVO(J)*(CYRHOO(J)-CYRHO(J))                          AIRBGG
      CYV(J)     = CYVMAX(J)*(1.0-Q2)                                   AIRBGG
      IF  (TIME.LT.CYTD(J))  GO TO 31                                   AIRBGG
      IF  (BD(1,JB).EQ.0.0)  GO TO 31                                   AIRBGG
      IF  (TIME.LE.0.0)      GO TO 31                                   AIRBGG
      VOLB = 0.0                                                        AIRBGG
C                                                                       AIRBGG
C     COMPUTE AIRBAG ELLIPSOID MATRIX AND ZERO BAG FORCE AND TORQUE.    AIRBGG
C                                                                       AIRBGG
      IF  (IFULL(J).NE.0)  GO TO 21                                     AIRBGG
      SAB = SCALE(J)*AB(1,J)                                            AIRBGG
      DO  19  I=1,3                                                     AIRBGG
   19 TMP(I) = DEPLOY(I,J) + SAB*DPVCTR(I,J)                            AIRBGG
      CALL DOT31 (D(1,1,NVEH),TMP,SEGLP(1,JB))                          AIRBGG
      DO  20  I=1,3                                                     AIRBGG
   20 SEGLP(I,JB) = SEGLP(I,JB) + SEGLP(I,NVEH)                         AIRBGG
   21 DO  23  I=1,3                                                     AIRBGG
      FORCE(I,J) = 0.0                                                  AIRBGG
   23 TORA (I,J) = 0.0                                                  AIRBGG
C                                                                       AIRBGG
C     COMPUTE FORCE,TORQUE AND VOLUME OF INTERSECTION                   AIRBGG
C     OF AIRBAG WITH REACTION PANEL ELLIPSOIDS.                         AIRBGG
C                                                                       AIRBGG
      KP = NPANEL(J)                                                    AIRBGG
      DO  26  K=1,KP                                                    AIRBGG
      CALL BGG(                                                         AIRBGG
     *   BD(7,JB),SEGLP(1,JB),D(1,1,JB),BD(4,JB),SEGLV(1,JB),WMEG(1,JB),AIRBGG
     *      B(1,K,J),SEGLP(1,NVEH),D(1,1,NVEH),BFB(1,K,J),SEGLV(1,NVEH),AIRBGG
     *      WMEG(1,NVEH),VSCS(J),IFULL(J),TV1(1,K,J),                   AIRBGG
     *      FRA(1,K),TORQ,TQB,VOLP(K,J))                                AIRBGG
      VOLBP(J) = VOLBP(J) + VOLP(K,J)                                   AIRBGG
      DO  26  I=1,3                                                     AIRBGG
      FORCE(I,J) = FORCE(I,J) + FRA(I,K)                                AIRBGG
   26 TORA (I,J) = TORA (I,J) + TORQ(I)                                 AIRBGG
C                                                                       AIRBGG
C     COMPUTE FORCE,TORQUE AND VOLUME OF INTERSECTION                   AIRBGG
C     OF AIRBAG WITH CONTACTING SEGMENT ELLIPSOIDS.                     AIRBGG
C                                                                       AIRBGG
      KBAG = MNBAG(J)                                                   AIRBGG
      DO  30  I=1,KBAG                                                  AIRBGG
      M  = MBAG(2,I,J)                                                  AIRBGG
      MM = MBAG(3,I,J)                                                  AIRBGG
      CALL BGG(                                                         AIRBGG
     *   BD(7,JB),SEGLP(1,JB),D(1,1,JB),BD(4,JB),SEGLV(1,JB),WMEG(1,JB),AIRBGG
     *      BD(7,MM),SEGLP(1,M),D(1,1,M),BD(4,MM),SEGLV(1,M),WMEG(1,M), AIRBGG
     *      VSCS(J),IFULL(J),TV2(1,I,J),FRB(1,I),TORQ,TQB(1,I),VOL(I))  AIRBGG
      IF  (VOL(I).EQ.0.0)  GO TO 30                                     AIRBGG
      VOLB = VOLB + VOL(I)                                              AIRBGG
      DO  28  K=1,3                                                     AIRBGG
      FORCE(K,J) = FORCE(K,J) + FRB(K,I)                                AIRBGG
```

```
28  TORA (K,J) = TORA (K,J) + TORQ(K)                                    AIRBGG
30  CONTINUE                                                             AIRBGG
    VOLBP(J) = VOLBP(J) + VOLB                                           AIRBGG
31  RETURN                                                               AIRBGG
    END                                                                  AIRBGG
```

```
      SUBROUTINE AIRBG1                                                 AIRBG1
C                                                REV IV    07/24/86SLIP
C     READS AND PRINTS THE INPUT CARDS THAT DESCRIBE THE PHYSICAL       AIRBG1
C     DIMENSIONS AND GAS DYNAMICS OF THE AIRBAG RESTRAINTS AND          AIRBG1
C     PERFORMS INITIALIZATION REQUIRED BY THE AIRBAG ROUTINE.           AIRBG1
C                                                                       AIRBG1
      IMPLICIT REAL*8 (A-H,O-Z)                                         AIRBG1
      COMMON/CONTRL/ TIME,NSEG,NJNT,NPL,NBLT,NBAG,NVEH,NGRND,           AIRBG1
     *               NS,NQ,NSD,NFLX,NHRNSS,NWINDF,NJNTF,NPRT(36),NPG    PAGE
      COMMON/SGMNTS/ D(3,3,30),WMEG(3,30),WMEGD(3,30),U1(3,30),U2(3,30),AIRBG1
     *               SEGLP(3,30),SEGLV(3,30),SEGLA(3,30),NSYM(30)       AIRBG1
      COMMON/DESCRP/ PHI(3,30),W(30),RW(30),SR(4,60),HA(3,60),HB(3,60), SLIP
     *               RPHI(3,30),HT(3,3,60),SPRING(5,90),VISC(7,90),     AIRBG1
     *               JNT(30),IPIN(30),ISING(30),IGLOB(30),JOINTF(30)    AIRBG1
      COMMON/FORCES/PSF(7,70),BSF(4,20),SSF(10,40),BAGSF(3,20),         NCFORC
     *               PRJNT(7,30),NPANEL(5),NPSF,NBSF,NSSF,NBGSF         AIRBG1
      COMMON/TITLES/ DATE(3),COMENT(40),VPSTTL(20),BDYTTL(5),           AIRBG1
     *               BLTTTL(5,8),PLTTL(5,30),BAGTTL(5,6),SEG(30),       AIRBG1
     *               JOINT(30),CGS(30),JS(30)                           AIRBG1
      REAL DATE,COMENT,VPSTTL,BDYTTL,BLTTTL,PLTTL,BAGTTL,SEG,JOINT      AIRBG1
      LOGICAL*1 CGS,JS                                                  AIRBG1
      COMMON/CNSNTS/ PI,RADIAN,G,THIRD,EPS(24),                         AIRBG1
     *               UNITL,UNITM,UNITT,GRAVTY(3),TWOPI                  TWOPI
      COMMON/CNTSRF/ PL(24,30),BELT(20,8),TPTS(6,8),BD(24,40)           EDGE
      COMMON/INTEST/ SGTEST(3,4,30),XTEST(3,120),SEGT(120),REGT(120)    AIRBG1
      REAL     SEGT                                                     AIRBG1
      COMMON/ABDATA/ ZDEP(3,5),DBR(3,3,5),DPVCTR(3,5),DEPLOY(3,5),      AIRBG1
     *               AB(3,5),B(9,4,5),ZR(3,4,5),BFB(3,4,5),DRR(9,4,5),  AIRBG1
     *               VBAGG(5),VSCS(5),SPRK(5),CK(5),CMASS(5),CYMIN(5),  AIRBG1
     *               CYMOUT(5),BAGPV(5),PD(5),VBAG(5),VOLBP(5),         AIRBG1
     *               PCYV(5),PCYMIN(5),PVBAG(5),TV1(3,4,5),TV2(3,10,5), AIRBG1
     *               SWITCH(5),PYMOUT(5),SCALE(5),PREVT,IFULL(6)        AIRBG1
      COMMON/CYDATA/ CYTD(5),CYPA(5),CYSP(5),CYT0(5),CYV0(5),CYCD(5),   AIRBG1
     *               CYK(5),CYR(5),CYAT(5),CYPV(5),CYCD0(5),CYA0(5),    AIRBG1
     *               CYP0(5),CYSS(5),CYL0(5),CYC(5),CYRHO0(5),CYVMAX(5),AIRBG1
     *               CYORFC(5),CYRHO(5),CYT(5),CYP(5),CYV(5)            AIRBG1
      COMMON/TEMPVS/ TMP(9),TMP1(3)                                     AIRBG1
      DIMENSION YB(3),YP(3),IDYPR(3)                                    AIRBG1
      REAL BAG(6)                                                       AIRBG1
      DATA BAG/4HBAG1,4HBAG2,4HBAG3,4HBAG4,4HBAG5,4HBAG /               AIRBG1
      DATA IDYPR/3,2,1/                                                 AIRBG1
      DATA MAXNPL/4/,MAXSEG/30/                                         CHGIII
C                                                                       AIRBG1
C     MAKE ROOM FOR BAG DATA IN SEGMENT ARRAYS BETWEEN VEH AND GRND.    AIRBG1
C                                                                       AIRBG1
      MSEG = 0                                                          CHGIII
      IF (NVEH.GT.NSEG) MSEG = NVEH - NSEG                              CHGIII
      L = NSEG + NBAG + MSEG + 1                                        CHGIII
      K = NSEG + MSEG + 1                                               CHGIII
      W(L)      = W(K)                                                  AIRBG1
```

```
      RW(L)      = RW(K)                                              AIRBG1
      SEG(L)     = SEG(K)                                             AIRBG1
      ISING(L)   = ISING(K)                                           AIRBG1
      IF  (L-1.GT.NJNT)  JNT (L-1) = 0                                AIRBG1
      IF  (L-1.GT.NJNT)  IPIN(L-1) = 0                                AIRBG1
      DO  19  I=1,3                                                   AIRBG1
      SEGLP(I,L) = SEGLP(I,K)                                         AIRBG1
      SEGLV(I,L) = SEGLV(I,K)                                         AIRBG1
      SEGLA(I,L) = SEGLA(I,K)                                         AIRBG1
      WMEG (I,L) = WMEG (I,K)                                         AIRBG1
      WMEGD(I,L) = WMEGD(I,K)                                         AIRBG1
      PHI  (I,L) = PHI  (I,K)                                         AIRBG1
      RPHI (I,L) = RPHI (I,K)                                         AIRBG1
      DO  18  J=1,3                                                   AIRBG1
      D(I,J,L) = D(I,J,K)                                             AIRBG1
   18 SGTEST(I,J,L) = SGTEST(I,J,K)                                   AIRBG1
   19 SGTEST(I,4,L) = SGTEST(I,4,K)                                   AIRBG1
      NGRND = NSEG + NBAG + MSEG + 1                                  CHGIII
      IF (NGRND.GT.MAXSEG) STOP 75                                    CHGIII
      DO 40 J=1,NBAG                                                  AIRBG1
      JB = NVEH + J                                                   AIRBG1
C                                                                     AIRBG1
C     READ AND PRINT CARDS D.4.A -D.4.F FOR THE JTH AIRBAG.           AIRBG1
C                                                                     AIRBG1
      READ(5,13) (BAGTTL(I,J),I = 1,5),NPANEL(J),                     AIRBG1
     *            (AB(I,J),I=1,3) , (BD(I,JB),I=4,6),                 AIRBG1
     *             YB,(ZDEP(I,J),I=1,3),                              AIRBG1
     *            W(JB),CYTD(J),CYPA(J),CYSP(J),CYTO(J),CYVO(J),      AIRBG1
     *            CYCD(J),CYK(J),CYR(J),CYAT(J),CYPV(J),CYCDO(J),     AIRBG1
     *             CYAO(J),SPRK(J),VSCS(J),CK(J),CMASS(J)             AIRBG1
   13 FORMAT (5A4,I4/(6F12.0))                                        AIRBG1
      IF (NPANEL(J).GT.MAXNPL) STOP 76                                CHGIII
      IF (MOD(J,2).EQ.1) WRITE(6,15) NPG                              PAGE
      IF (MOD(J,2).EQ.1) NPG=NPG+1                                    PAGE
   15 FORMAT('1',122X,'PAGE',I5/'  AIRBAG INPUTS',105X,'CARDS D.4')   PAGE
      WRITE(6,14) J,(BAGTTL(I,J),I = 1,5 ),                           AIRBG1
     *            (AB(I,J),I=1,3) , (BD(I,JB),I=4,6),                 AIRBG1
     *             YB,(ZDEP(I,J),I=1,3),                              AIRBG1
     *            W(JB),CYTD(J),CYPA(J),CYSP(J),CYTO(J),CYVO(J),      AIRBG1
     *            CYCD(J),CYK(J),CYR(J),CYAT(J),CYPV(J),CYCDO(J),     AIRBG1
     *             CYAO(J),SPRK(J),VSCS(J),CK(J),CMASS(J)             AIRBG1
   14 FORMAT('0 AIRBAG NO.',I4,4X,5A4//                               AIRBG1
     * 29X,'AIR BAG SEMIAXES',46X,'C.G. OFFSET'/6X,6G20.9//           AIRBG1
     * 15X,'YAW',16X,'PITCH',15X,'ROLL',30X,'DEPLOYMENT POINT'        AIRBG1
     *                                            /6X,6G20.9//        AIRBG1
     * 15X,'XBM',16X,'CYTD',16X,'CYPA',16X,'CYSP',16X,'CYTO',16X,'CYVO'  AIRBG1
     *                                            /6X,6G20.9//        AIRBG1
     * 14X,'CYCD',17X,'CYK',17X,'CYR',16X,'CYAT',16X,'CYPV',16X,'CYCDO'  AIRBG1
     *                                            /6X,6G20.9//        AIRBG1
     *14X,'CYAO',16X,'SPRK',16X,'VSCS',17X,'CK',17X,'CMASS'/6X,5G20.9)   AIRBG1
```

```
      KP = NPANEL(J)                                                    AIRBG1
      DO  25  K=1,KP                                                    AIRBG1
C                                                                       AIRBG1
C     READ AND PRINT CARDS D.4.G AND D.4.H FOR THE KTH PANEL TO         AIRBG1
C     CONTACT THE JTH AIRBAG. THESE PANELS ARE APPROXIMATED BY          AIRBG1
C     ELLIPSOIDS. THE FIRST PANEL (K=1) IS THE REACTION PANEL THAT      AIRBG1
C     INCLUDES THE DEPLOYMENT POINT.                                    AIRBG1
C                                                                       AIRBG1
      READ(5,11) (B(I,K,J),I=1,3),(BFB(I,K,J),I=1,3),                   AIRBG1
     *           (ZR(I,K,J),I=1,3),YP                                   AIRBG1
   11 FORMAT(6F12.0)                                                    AIRBG1
      WRITE(6,12) K,(B(I,K,J),I=1,3),(BFB(I,K,J),I=1,3),                AIRBG1
     *             (ZR(I,K,J),I=1,3),YP                                 AIRBG1
   12 FORMAT('0 PANEL NO.',I4//                                         AIRBG1
     * 24X,'PANEL ELLIPSOID SEMIAXES',43X,'C.G. OFFSET'/6X,6G20.9//     AIRBG1
     * 29X,'PANEL LOCATION',32X,'YAW',16X,'PITCH',15X,'ROLL'/6X,6G20.9) AIRBG1
C                                                                       AIRBG1
C     CONVERT B FROM ELLIPSOID SEMIAXES TO MATRIX                       AIRBG1
C                                                                       AIRBG1
      DO 21 I=1,3                                                       AIRBG1
   21 TMP(I) = B(I,K,J)                                                 AIRBG1
      DO 22 I=1,9                                                       AIRBG1
   22 B(I,K,J) = 0.0                                                    AIRBG1
      DO 23 I=1,3                                                       AIRBG1
   23 B(4*I-3,K,J) = 1.0/TMP(I)**2                                      AIRBG1
      CALL DRCYPR (DRR(1,K,J),YP,IDYPR)                                 AIRBG1
      CALL MAT33 (B(1,K,J),DRR(1,K,J),TMP)                              AIRBG1
      CALL DOT33 (DRR(1,K,J),TMP,B(1,K,J))                              AIRBG1
      CALL DOT31 (DRR(1,K,J),BFB(1,K,J),TMP)                            AIRBG1
      DO  24  I=1,3                                                     AIRBG1
 24   BFB(I,K,J) = TMP(I) + ZR(I,K,J)                                   AIRBG1
 25   CONTINUE                                                          AIRBG1
C                                                                       AIRBG1
C     COMPUTE GEOMETRY OF DEPLOYMENT POINT ON FIRST PANEL.              AIRBG1
C                                                                       AIRBG1
      CALL DRCYPR (DBR(1,1,J),YB,IDYPR)                                 AIRBG1
      CALL DOT31 (DRR(1,1,J),ZDEP(1,J),DEPLOY(1,J))                     AIRBG1
      DO  31  I=1,3                                                     AIRBG1
      DPVCTR(I,J) = -DBR(1,I,J)                                         AIRBG1
 31   DEPLOY(I,J) = DEPLOY(I,J) + BFB(I,1,J)                            AIRBG1
      CALL PANEL (DBR(1,1,J),DEPLOY(1,J),JB)                            AIRBG1
C                                                                       AIRBG1
C     INITIALIZATION OF AIRBAG GEOMETRY.                                AIRBG1
C                                                                       AIRBG1
      VBAGG(J) = 4.0/3.0*PI*AB(1,J)*AB(2,J)*AB(3,J)                     AIRBG1
      PHI(1,JB) = (AB(2,J)**2+AB(3,J)**2)/5.0                           AIRBG1
      PHI(2,JB) = (AB(3,J)**2+AB(1,J)**2)/5.0                           AIRBG1
      PHI(3,JB) = (AB(1,J)**2+AB(2,J)**2)/5.0                           AIRBG1
      JNT(JB-1) = 0                                                     AIRBG1
      IPIN(JB-1) = 0                                                    AIRBG1
```

```
      SEG(JB) = BAG(J)                                                  AIRBG1
      IF  (NBAG.EQ.1)  SEG(JB) = BAG(6)                                 AIRBG1
      ISING(JB) = -1                                                    AIRBG1
      RW(JB) = G/W(JB)                                                  AIRBG1
      DO  36  I=1,3                                                     AIRBG1
      BD(I,JB) = 0.0                                                    AIRBG1
      RPHI(I,JB) = 1.0/PHI(I,JB)                                        AIRBG1
      DO  36  K=1,4                                                     AIRBG1
   36 SGTEST(I,K,JB) = 0.0                                              AIRBG1
      DO  35  I=7,24                                                    AIRBG1
   35 BD(I,JB) = 0.0                                                    AIRBG1
      IFULL(J) = 0                                                      AIRBG1
      CYMOUT(J) = 0.0                                                   AIRBG1
      PYMOUT(J) = 0.0                                                   AIRBG1
      DO  38  I=1,3                                                     AIRBG1
      DO  37  K=1,4                                                     AIRBG1
   37 TV1(I,K,J) = 0.0                                                  AIRBG1
      DO  38  K=1,10                                                    AIRBG1
   38 TV2(I,K,J) = 0.0                                                  AIRBG1
C                                                                       AIRBG1
C     AIR CYLINDER INITIALIZATION                                       AIRBG1
C                                                                       AIRBG1
      CYPO(J)  = CYSP(J)+CYPA(J)                                        AIRBG1
      CYSS(J)  = DSQRT(CYK(J)*CYR(J)*CYTO(J)*G)                         AIRBG1
      CYLO(J)  = CYVO(J)/CYAT(J)                                        AIRBG1
      CYK1     = CYK(J)-1.0                                             AIRBG1
      CYK2     = 0.5*(CYK(J)+1.0)                                       AIRBG1
      CYK3     = CYK2**(-CYK2/CYK1)                                     AIRBG1
      CYC(J)   = 0.5*CYK1*CYSS(J)*CYCD(J)/CYLO(J)*CYK3                  AIRBG1
      CYRHOO(J) = CYPO(J)/(CYR(J)*CYTO(J))                              AIRBG1
      CYVMAX(J) = CYVO(J)/CYK(J)*CYPO(J)/CYPA(J)                        AIRBG1
      CYORFC(J) = CYCDO(J)*CYAO(J)*G*DSQRT(2.0*CYPA(J)*CYK(J))/CYSS(J)  AIRBG1
      IF (NPRT(22).NE.0) WRITE(6,39)                                    AIRBG1
     *     (SEGLP(I,JB),I=1,3),(SEGLV(I,JB),I=1,3),(WMEG(I,JB),I=1,3),  AIRBG1
     *     VBAGG(J),CYPO(J),CYSS(J),CYC(J),CYRHOO(J),CYVMAX(J),CYORFC(J)AIRBG1
   39 FORMAT('0 AIRBAG SINPUT'/(1X,9G14.6))                             AIRBG1
   40 CONTINUE                                                          AIRBG1
      PREVT = 0.0                                                       AIRBG1
      RETURN                                                            AIRBG1
      END                                                               AIRBG1
```

```
      SUBROUTINE AIRBG3(IRESET)                                         AIRBG3
C                                                    REV IV    07/23/86TWOPI
C                                                                       AIRBG3
C     THIS SUBROUTINE IS CALLED BY SUBROUTINE UPDATE AT START (IRESET=1)AIRBG3
C     AND END (IRESET=2) OF EACH INTEGRATION STEP TO DETERMINE IF EACH  AIRBG3
C     AIRBAG HAS BEEN FULLY INFLATED.                                   AIRBG3
C                                                                       AIRBG3
      IMPLICIT REAL*8 (A-H,O-Z)                                         AIRBG3
      COMMON/CONTRL/ TIME,NSEG,NJNT,NPL,NBLT,NBAG,NVEH,NGRND,           AIRBG3
     *               NS,NQ,NSD,NFLX,NHRNSS,NWINDF,NJNTF,NPRT(36),NPG    PAGE
      COMMON/SGMNTS/ D(3,3,30),WMEG(3,30),WMEGD(3,30),U1(3,30),U2(3,30),AIRBG3
     *               SEGLP(3,30),SEGLV(3,30),SEGLA(3,30),NSYM(30)       AIRBG3
      COMMON/JBARTZ/ MNPL(   30),MNBLT(   8),MNSEG(   30),MNBAG(    6), AIRBG3
     *               MPL(3,5,30),MBLT(3,5,8),MSEG(3,5,30),MBAG(3,10,6), AIRBG3
     *               NTPL( 5,30),NTBLT( 5,8),NTSEG( 5,30)               AIRBG3
      COMMON/FORCES/PSF(7,70),BSF(4,20),SSF(10,40),BAGSF(3,20),         NCFORC
     *               PRJNT(7,30),NPANEL(5),NPSF,NBSF,NSSF,NBGSF         AIRBG3
      COMMON/CNTSRF/ PL(24,30),BELT(20,8),TPTS(6,8),BD(24,40)           EDGE
      COMMON/CNSNTS/ PI,RADIAN,G,THIRD,EPS(24),                         AIRBG3
     *               UNITL,UNITM,UNITT,GRAVTY(3),TWOPI                  TWOPI
      COMMON/ABDATA/ ZDEP(3,5),DBR(3,3,5),DPVCTR(3,5),DEPLOY(3,5),      AIRBG3
     *               AB(3,5),B(9,4,5),ZR(3,4,5),BFB(3,4,5),DRR(9,4,5),  AIRBG3
     *               VBAGG(5),VSCS(5),SPRK(5),CK(5),CMASS(5),CYMIN(5),  AIRBG3
     *               CYMOUT(5),BAGPV(5),PD(5),VBAG(5),VOLBP(5),         AIRBG3
     *               PCYV(5),PCYMIN(5),PVBAG(5),TV1(3,4,5),TV2(3,10,5), AIRBG3
     *               SWITCH(5),PYMOUT(5),SCALE(5),PREVT,IFULL(6)        AIRBG3
      COMMON/CYDATA/ CYTD(5),CYPA(5),CYSP(5),CYTO(5),CYVO(5),CYCD(5),   AIRBG3
     *               CYK(5),CYR(5),CYAT(5),CYPV(5),CYCDO(5),CYAO(5),    AIRBG3
     *               CYPO(5),CYSS(5),CYLO(5),CYC(5),CYRHOO(5),CYVMAX(5),AIRBG3
     *               CYORFC(5),CYRHO(5),CYT(5),CYP(5),CYV(5)            AIRBG3
      COMMON/TEMPVS/ TMP(9),TMP1(3)                                     AIRBG3
      CALL ELTIME(1,29)                                                 AIRBG3
      JRESET = IRESET                                                   AIRBG3
      IF  (JRESET.EQ.1)  PREVT = TIME                                   AIRBG3
      NBGSF = 0                                                         AIRBG3
      DO  50  J=1,NBAG                                                  AIRBG3
      IF  (MNBAG(J).EQ.0)  GO TO 50                                     AIRBG3
      JB = NVEH + J                                                     AIRBG3
      JFULL = IFULL(J) + 2                                              AIRBG3
      IF  (JFULL.LT.1 .OR. JFULL.GT.3)  GO TO 11                        AIRBG3
      IF (JRESET-1) 13,13,14                                            BUTLER1
   11 WRITE(6,12)  TIME                                                 AIRBG3
   12 FORMAT ('0 ERROR IN SUBROUTINE AIRBG3 AT TIME =',F10.6)           AIRBG3
      STOP 32                                                           AIRBG3
   13 IF (JFULL-2) 41,49,49                                             BUTLER1
   14 IF (JFULL-2) 11,21,31                                             BUTLER1
C                                                                       AIRBG3
C     END OF INTEGRATION STEP WHEN IFULL=0. TEST FOR FULL INFLATION.    AIRBG3
C                                                                       AIRBG3
   21 PD(J) = 0.0                                                       AIRBG3
```

```
      PCYV(J) = CYV(J)                                                  AIRBG3
      PCYMIN(J) = CYMIN(J)                                              AIRBG3
      PVBAG(J) = VBAG(J)                                                AIRBG3
   22 CALL AIRBGG(J)                                                    AIRBG3
      VBAG(J) = CYV(J) + VOLBP(J)                                       AIRBG3
      IF  (SCALE(J).EQ.1.0)  GO TO 23                                   AIRBG3
      SCALE(J) = (VBAG(J)/VBAGG(J))**THIRD                              AIRBG3
      IF  (SCALE(J).LT.1.0)  GO TO 24                                   AIRBG3
      SCALE(J) = 1.0                                                    AIRBG3
      GO TO 22                                                          AIRBG3
   23 IFULL(J) = -1                                                     AIRBG3
      CYMOUT(J) = 0.0                                                   AIRBG3
      PSW1 = (VBAG (J)-VBAGG(J))*PCYV(J)/PCYMIN(J)                      AIRBG3
      PSW2 = (VBAGG(J)-PVBAG(J))* CYV(J)/ CYMIN(J)                      AIRBG3
      SWITCH(J) = (PSW1+PSW2)/(VBAG(J)-PVBAG(J))                        AIRBG3
      BAGPV(J) = CYPA(J)*(CYMIN(J)*SWITCH(J))**CYK(J)                   AIRBG3
      PD(J) = BAGPV(J)/(CYV(J)**CYK(J)) - CYPA(J)                       AIRBG3
   24 DO  25  K=1,3                                                     AIRBG3
      BD(K,JB) = SCALE(J)*AB(K,J)                                       AIRBG3
      IF  (SCALE(J).EQ.0.0)  GO TO 25                                   AIRBG3
      BD(4*K+12,JB) = BD(K,JB)**2                                       AIRBG3
      BD(4*K+ 3,JB) = 1.0/BD(4*K+12,JB)                                 AIRBG3
   25 TMP(K) = DEPLOY(K,J) + BD(1,JB)*DPVCTR(K,J)                       AIRBG3
      CALL PANEL (DBR(1,1,J),TMP,JB)                                    AIRBG3
C                                                                       AIRBG3
C     SET UP BAGSF ARRAY FOR OUTPUT.                                    AIRBG3
C                                                                       AIRBG3
   31 BAGSF(1,NBGSF+1) = CYP(J)                                         AIRBG3
      BAGSF(2,NBGSF+1) = CYT(J)                                         AIRBG3
      BAGSF(3,NBGSF+1) =  PD(J)                                         AIRBG3
      CALL DOT31 (D(1,1,JB),BD(4,JB),TMP)                               AIRBG3
      DO  32  K=1,3                                                     AIRBG3
      BAGSF(K,NBGSF+3) = BD(K,JB)                                       AIRBG3
   32 TMP(K) = TMP(K) + SEGLP(K,JB) - SEGLP(K,NVEH)                     AIRBG3
      CALL MAT31 (D(1,1,NVEH),TMP,BAGSF(1,NBGSF+2))                     AIRBG3
      CALL YPRDEG (D(1,1,JB),BAGSF(1,NBGSF+4))                          AIRBG3
      NBGSF = NBGSF + 5 + NPANEL(J) + MNBAG(J)                          AIRBG3
      GO TO 50                                                          AIRBG3
C                                                                       AIRBG3
C     START OF INTEGRATION STEP WITH IFULL = -1, RESET INTEGRATOR.      AIRBG3
C                                                                       AIRBG3
   41 IFULL(J) = 1                                                      AIRBG3
      IRESET = -1                                                       AIRBG3
   49 PYMOUT(J) = CYMOUT(J)                                             AIRBG3
   50 CONTINUE                                                          AIRBG3
      CALL ELTIME(2,29)                                                 AIRBG3
      RETURN                                                            AIRBG3
      END                                                               AIRBG3
```

```
      SUBROUTINE BELTG (ZA,ZB,ZC,BD)                                    BELTG
C                                                  REV IV    07/23/86TWOPI
C     COMPUTE TANGENT POINTS, UNIT VECTORS FROM TANGENT POINTS TO       BELTG
C     ANCHOR POINTS AND LENGTHS OF THE BELT SEGMENTS.                   BELTG
C                                                                       BELTG
C     ARGUMENTS:                                                        BELTG
C                                                                       BELTG
C         ZA,ZB - ANCHOR POINTS RELATIVE TO ELLIPSOID CENTER.           BELTG
C         ZC    - FIXED POINT OF BELT ON SEGMENT ELLIPSOID.             BELTG
C         BD    - SEGMENT ELLIPSOID SEMIAXES AND CENTER.                BELTG
C                                                                       BELTG
C     RESULTS ARE RETURNED TO CALLING ROUTINE VIA COMMON/TEMPVS/.       BELTG
C                                                                       BELTG
      IMPLICIT REAL*8 (A-H,O-Z)                                         BELTG
      DIMENSION ZA(3),ZB(3),ZC(3),BD(24)                                BELTG
      COMMON/CONTRL/ TIME,NSEG,NJNT,NPL,NBLT,NBAG,NVEH,NGRND,           BELTG
     *               NS,NQ,NSD,NFLX,NHRNSS,NWINDF,NJNTF,NPRT(36),NPG    PAGE
      COMMON/CNSNTS/ PI,RADIAN,G,THIRD,EPS(24),                         BELTG
     *               UNITL,UNITM,UNITT,GRAVTY(3),TWOPI                  TWOPI
C     NOTE: BELTRT AND BELTG SHARE FIRST PART OF TEMPVS                 BELTG
      COMMON/TEMPVS/ APA(3),UVA(3),DLGA,UAA,APB(3),UVB(3),DLGB,UBB      BELTG
     *              ,TA(3),TB(3),TC(3),UP(3),B(3)                       BELTG
     *              ,UC(3),AX(3),XE(3),BX(3),ACA(3),ACB(3)              BELTG
C                                                                       BELTG
C     COMPUTE                                                           BELTG
C       TC: NORMALIZED VECTOR OF BELT PLANE DETERMINED                  BELTG
C           BY ANCHOR POINTS AND FIXED POINT.                           BELTG
C                                                                       BELTG
      DO 10 K=1,3                                                       BELTG
      TA(K) = ZC(K)-ZA(K)                                               BELTG
   10 TB(K) = ZC(K)-ZB(K)                                               BELTG
      CALL CROSS(TB,TA,TC)                                              BELTG
      S = DSQRT(TC(1)**2 + TC(2)**2 + TC(3)**2)                         BELTG
      TC(1) = TC(1)/S                                                   BELTG
      TC(2) = TC(2)/S                                                   BELTG
      TC(3) = TC(3)/S                                                   BELTG
C                                                                       BELTG
C     GET DISTANCE OF BELT PLANE TO CENTER OF ELLIPSIOD.                BELTG
C                                                                       BELTG
      BET = TC(1)*ZC(1)+TC(2)*ZC(2)+TC(3)*ZC(3)                         BELTG
C                                                                       BELTG
C     COMPUTE                                                           BELTG
C       XE: CENTER OF ELLIPSE DETERMINED BY INTERSECTION                BELTG
C           OF BELT PLANE AND SEGMENT ELLIPSOID.                        BELTG
C                                                                       BELTG
      CALL MAT31 (BD(16),TC,XE)                                         BELTG
      GG = BET/(TC(1)*XE(1)+TC(2)*XE(2)+TC(3)*XE(3))                    BELTG
      DLGA = 0.0                                                        BELTG
      DLGB = 0.0                                                        BELTG
      DO 15 K=1,3                                                       BELTG
```

```
      XE(K)   = XE(K)*GG                                                 BELTG
      UC(K)   = ZC(K)-XE(K)                                              BELTG
      APA(K)  = UC(K)                                                    BELTG
   15 APB(K)  = UC(K)                                                    BELTG
      YAY = GG*BET                                                       BELTG
      YAY1 = 1.0-YAY                                                     BELTG
      IF (YAY1.LE.EPS(6)) GO TO 70                                       BELTG
C                                                                        BELTG
C     CALCULATE POSSIBLE TANGENT POINTS FROM                             BELTG
C       UVA,UVB: VECTORS FROM ELLIPSE CENTER TO MIDPOINT OF              BELTG
C                LINE CONNECTING POSSIBLE TANGENT POINTS.                BELTG
C       ACA,ACB: VECTORS FROM UVA,UVB TO TANGENT POINTS (POSITIVE).     BELTG
C                                                                        BELTG
      CALL MAT31 (BD(7),ZA,AX)                                           BELTG
      CALL MAT31 (BD(7),ZB,BX)                                           BELTG
      ZZA = AX(1)*ZA(1)+AX(2)*ZA(2)+AX(3)*ZA(3)                          BELTG
      IF( ZZA.LE.1.0) STOP 88                                            CHGIII
      ZZB = BX(1)*ZB(1)+BX(2)*ZB(2)+BX(3)*ZB(3)                          BELTG
      IF( ZZB.LE.1.0) STOP 89                                            CHGIII
      C2A = YAY1/(ZZA-YAY)                                               BELTG
      C2B = YAY1/(ZZB-YAY)                                               BELTG
      CALL CROSS(TC,AX,ACA)                                              BELTG
      CALL CROSS(TC,BX,ACB)                                              BELTG
      TTA = 0.0                                                          BELTG
      TTB = 0.0                                                          BELTG
      DO 21 I=1,3                                                        BELTG
      DO 21 J=1,3                                                        BELTG
      K = 3*J+I+3                                                        BELTG
      TTA = TTA + ACA(I)*BD(K)*ACA(J)                                    BELTG
   21 TTB = TTB + ACB(I)*BD(K)*ACB(J)                                    BELTG
      C3A = DSQRT((1.0 - C2A)*YAY1/TTA)                                  CHGIII
      C3B = DSQRT((1.0 - C2B)*YAY1/TTB)                                  CHGIII
      TT = DSQRT(UC(1)**2 + UC(2)**2 + UC(3)**2)                         BELTG
      DO 24 K=1,3                                                        BELTG
      UVA(K) = C2A*(ZA(K)-XE(K))                                         BELTG
      UVB(K) = C2B*(ZB(K)-XE(K))                                         BELTG
      ACA(K) = C3A*ACA(K)                                                BELTG
      ACB(K) = C3B*ACB(K)                                                BELTG
      UC(K) = UC(K)/TT                                                   BELTG
   24 B(K) = 0.0                                                         BELTG
C                                                                        BELTG
C     OBTAIN EQUATION OF ELLIPSE                                         BELTG
C        B1*X**2 + 2*B2*X*Y + B3*Y**2 = 1                                BELTG
C     IN UC,UP COORDINATES WHERE UC POINTS TO FIXED POINT.               BELTG
C                                                                        BELTG
      CALL CROSS(TC,UC,UP)                                               BELTG
      DO 22 I=1,3                                                        BELTG
      DO 22 J=1,3                                                        BELTG
      K = 3*J+I+3                                                        BELTG
      B(1) = B(1) + UC(I)*BD(K)*UC(J)                                    BELTG
```

```
      B(2) = B(2) + UC(I)*BD(K)*UP(J)                                   BELTG
   22 B(3) = B(3) + UP(I)*BD(K)*UP(J)                                   BELTG
      B(1) = B(1)/YAY1                                                  BELTG
      B(2) = B(2)/YAY1                                                  BELTG
      B(3) = B(3)/YAY1                                                  BELTG
C                                                                       BELTG
C     COMPUTE ANGLES FROM FIXED POINT TO POSSIBLE TANGENT POINTS.       BELTG
C                                                                       BELTG
      UCUVA = UC(1)*UVA(1) + UC(2)*UVA(2) + UC(3)*UVA(3)                BELTG
      UCUVB = UC(1)*UVB(1) + UC(2)*UVB(2) + UC(3)*UVB(3)                BELTG
      UCACA = UC(1)*ACA(1) + UC(2)*ACA(2) + UC(3)*ACA(3)                BELTG
      UCACB = UC(1)*ACB(1) + UC(2)*ACB(2) + UC(3)*ACB(3)                BELTG
      UPUVA = UP(1)*UVA(1) + UP(2)*UVA(2) + UP(3)*UVA(3)                BELTG
      UPUVB = UP(1)*UVB(1) + UP(2)*UVB(2) + UP(3)*UVB(3)                BELTG
      UPACA = UP(1)*ACA(1) + UP(2)*ACA(2) + UP(3)*ACA(3)                BELTG
      UPACB = UP(1)*ACB(1) + UP(2)*ACB(2) + UP(3)*ACB(3)                BELTG
      TH1 = DATAN2(UPUVA-UPACA,UCUVA-UCACA)                             BELTG
      TH2 = DATAN2(UPUVA+UPACA,UCUVA+UCACA)                             BELTG
      TH3 = DATAN2(UPUVB-UPACB,UCUVB-UCACB)                             BELTG
      TH4 = DATAN2(UPUVB+UPACB,UCUVB+UCACB)                             BELTG
      IF (TH1.LT.0.0) TH1 = TWOPI + TH1                                 BELTG
      IF (TH2.LT.0.0) TH2 = TWOPI + TH2                                 BELTG
      IF (TH3.LT.0.0) TH3 = TWOPI + TH3                                 BELTG
      IF (TH4.LT.0.0) TH4 = TWOPI + TH4                                 BELTG
C                                                                       BELTG
C     CHOOSE PROPER TANGENT POINTS AND BELT ARC LENGTHS.                BELTG
C                                                                       BELTG
      THMIN = DMIN1(TH1,TH2,TH3,TH4)                                    BELTG
      IF (THMIN.EQ.TH1.AND.DMIN1(TH2,TH3,TH4).NE.TH4) GO TO 61          BELTG
      IF (THMIN.EQ.TH2.AND.DMAX1(TH1,TH3,TH4).EQ.TH4) GO TO 61          BELTG
      IF (THMIN.EQ.TH3.AND.DMIN1(TH1,TH2,TH4).NE.TH2) GO TO 63          BELTG
      IF (THMIN.EQ.TH4.AND.DMAX1(TH1,TH2,TH3).EQ.TH2) GO TO 63          BELTG
      GO TO 70                                                          BELTG
   61 THA = TH1                                                         BELTG
      THB = TWOPI-TH4                                                   BELTG
      DO 62 K=1,3                                                       BELTG
      APA(K) = UVA(K)-ACA(K)                                            BELTG
   62 APB(K) = UVB(K)+ACB(K)                                            BELTG
      GO TO 65                                                          BELTG
   63 THA = TWOPI-TH2                                                   BELTG
      THB = TH3                                                         BELTG
      DO 64 K=1,3                                                       BELTG
      APA(K) = UVA(K)+ACA(K)                                            BELTG
   64 APB(K) = UVB(K)-ACB(K)                                            BELTG
   65 CONTINUE                                                          BELTG
      EPS1 = EPS(1)                                                     BELTG
      DLGA = DABS(ELONG(B(1),B(2),B(3),EPS1,THA))                       BELTG
      DLGB = DABS(ELONG(B(1),B(2),B(3),EPS1,THB))                       BELTG
C                                                                       BELTG
C     CALCULATE BELT LENGTHS AND UNIT VECTORS                           BELTG
```

```
C     FROM TANGENT POINTS TO ANCHOR POINTS.                            BELTG
C                                                                      BELTG
   70 UAA=0.                                                           BELTG
      UBB=0.                                                           BELTG
      DO 80 K=1,3                                                      BELTG
      APA(K) = APA(K)+XE(K)                                            BELTG
      APB(K) = APB(K)+XE(K)                                            BELTG
      UVA(K)=ZA(K)-APA(K)                                              BELTG
      UVB(K)=ZB(K)-APB(K)                                              BELTG
      APA(K)=APA(K)+BD(K+3)                                            BELTG
      APB(K)=APB(K)+BD(K+3)                                            BELTG
      UAA=UAA+UVA(K)**2                                                BELTG
      UBB=UBB+UVB(K)**2                                                BELTG
   80 CONTINUE                                                         BELTG
      UAA=DSQRT(UAA)                                                   BELTG
      UBB=DSQRT(UBB)                                                   BELTG
      DO 90 K=1,3                                                      BELTG
      UVA(K)=UVA(K)/UAA                                                BELTG
      UVB(K)=UVB(K)/UBB                                                BELTG
   90 CONTINUE                                                         BELTG
C                                                                      BELTG
C     OPTIONAL OUTPUT                                                  BELTG
C                                                                      BELTG
      IF (NPRT(15).EQ.0) GO TO 99                                      BELTG
      WRITE(6,50)                                                      BELTG
   50 FORMAT(1X,'BELT RESTRAINT')                                      BELTG
      WRITE(6,60) APA,UVA,DLGA,UAA                                     BELTG
      WRITE(6,60) APB,UVB,DLGB,UBB                                     BELTG
   60 FORMAT (1X,1P8D15.5)                                             BELTG
   99 CONTINUE                                                         BELTG
      RETURN                                                           BELTG
      END                                                              BELTG
```

```
      SUBROUTINE BELTRT(I,II,MM,M,NT)                                   BELTRT
C                                                   REV IV    07/23/86TWOPI
C     THE ROUTINE CALLS SUBROUTINE BELTG TO COMPUTE THE TANGENT POINTS  BELTRT
C     AND BELT LENGTHS AND APPLIES THE RESTRAINT FORCES TO THE U1 ARRAY BELTRT
C     AND BELT TORQUES TO THE U2 ARRAY FOR ELLIPSOID(II) ATTACHED TO    BELTRT
C     BODY SEGMENT (I) BY BELT (M) ATTACHED TO SEGMENT (MM).            BELTRT
C                                                                       BELTRT
      IMPLICIT REAL*8(A-H,O-Z)                                          BELTRT
      COMMON/CONTRL/ TIME,NSEG,NJNT,NPL,NBLT,NBAG,NVEH,NGRND,           BELTRT
     *               NS,NQ,NSD,NFLX,NHRNSS,NWINDF,NJNTF,NPRT(36),NPG    PAGE
      COMMON/SGMNTS/ D(3,3,30),WMEG(3,30),WMEGD(3,30),U1(3,30),U2(3,30),BELTRT
     *               SEGLP(3,30),SEGLV(3,30),SEGLA(3,30),NSYM(30)       BELTRT
      COMMON/CNTSRF/ PL(24,30),BELT(20,8),TPTS(6,8),BD(24,40)           EDGE
      COMMON/TABLES/MXNTI,MXNTB,MXTB1,MXTB2,NTI(50),NTAB(1250),TAB(4500)DIMENB
      COMMON/FORCES/PSF(7,70),BSF(4,20),SSF(10,40),BAGSF(3,20),         NCFORC
     *               PRJNT(7,30),NPANEL(5),NPSF,NBSF,NSSF,NBGSF         BELTRT
      COMMON/CNSNTS/ PI,RADIAN,G,THIRD,EPS(24),                         BELTRT
     *               UNITL,UNITM,UNITT,GRAVTY(3),TWOPI                  TWOPI
      COMMON/RSAVE/  XSG(3,20,3),DPMI(3,3,30),LPMI(30),                 TGMOD4
     *               NSG(9),MSG(20,9),MCG,MCGIN(24,5),KREF(20,9)        TTHKREF
C     NOTE: BELTRT AND BELTG SHARE FIRST PART OF TEMPVS                 BELTRT
      COMMON/TEMPVS/ APA(3),UVA(3),DLGA,UAA,APB(3),UVB(3),DLGB,UBB      BELTRT
      DIMENSION TA(3),TB(3),ZA(3),ZB(3),TT(3),TTT(3),TA1(3),TB1(3)      TGMOD4
C                                                                       BELTRT
      CALL ELTIME(1,22)                                                 BELTRT
C                                                                       BELTRT
C     CONVERT SEGMENT POSITION TO SEGMENT REFERENCE.                    BELTRT
C                                                                       BELTRT
      MA = MOD(MM,100)                                                  JTF984
      MB = MM/100                                                       JTF984
      IF (MB.EQ.0) MB=MA                                                JTF984
      CALL DOT31 (D(1,1,MA),BELT(1,M),TA)                               BELTRT
      CALL DOT31 (D(1,1,MB),BELT(4,M),TB)                               BELTRT
      DO  10  K=1,3                                                     BELTRT
      TA(K) = SEGLP(K,MA) + TA(K) - SEGLP(K,I)                          BELTRT
   10 TB(K) = SEGLP(K,MB) + TB(K) - SEGLP(K,I)                          BELTRT
      CALL MAT31 (D(1,1,I),TA,ZA)                                       BELTRT
      CALL MAT31 (D(1,1,I),TB,ZB)                                       BELTRT
      DO  13  K=1,3                                                     BELTRT
      ZA(K) = ZA(K) - BD(K+3,II)                                        BELTRT
   13 ZB(K) = ZB(K) - BD(K+3,II)                                        BELTRT
C                                                                       BELTRT
C     COMPUTE NEW BELT LENGTHS AND EXPANSION.                           BELTRT
C                                                                       BELTRT
      CALL BELTG (ZA, ZB, BELT(7,M), BD(1,II))                          BELTRT
      TLA = DLGA+UAA                                                    BELTRT
      TLB = DLGB+UBB                                                    BELTRT
      TL  = TLA+TLB                                                     BELTRT
      IF  (TIME.NE.0.0)  GO TO 11                                       BELTRT
C                                                                       BELTRT
```

```
C     IF TIME=0, COMPUTE INITIAL BELT LENGTHS                               BELTRT
C     AND STORE RESULTS IN BELT ARRAY.                                      BELTRT
C                                                                           BELTRT
      IF (BELT(11,M).LT.0.0) BELT(11,M)= -BELT(11,M)-TL                     BELTRT
      IF (BELT(11,M).LT.0.0) BELT(11,M)=0.0                                 BELTRT
      BELT(12,M) = TLA+TLA/TL*BELT(11,M)                                    BELTRT
      BELT(13,M) = TLB+TLB/TL*BELT(11,M)                                    BELTRT
      B1213 = BELT(12,M) + BELT(13,M)                                       BELTRT
      BELT(10,M) = B1213                                                    BELTRT
      DO 305 LL=1,3                                                         TGMOD4
      TA1(LL) = APA(LL)                                                     TGMOD4
  305 TB1(LL) = APB(LL)                                                     TGMOD4
      IF(LPMI(I).EQ.0) GO TO 306                                            TGMOD4
      CALL DOT31(DPMI(1,1,I),APA,TA1)                                       TGMOD4
      CALL DOT31(DPMI(1,1,I),APB,TB1)                                       TGMOD4
  306 CONTINUE                                                              TGMOD4
      WRITE (6,14) M, B1213, BELT(12,M), BELT(13,M), UNITL,I,TA1, TB1       TGMOD4
   14 FORMAT('0 INITIAL LENGTHS OF BELT NO.',I3,' AND ITS SEGMENTS ARE',BELTRT
     *        3F12.4,1X,A4/'0 INITIAL TANGENT POINTS IN LOCAL REFERENCE TGMOD4
     *OF SEGMENT ',I2,' ARE:',/,2(3X,3F12.3))                              TGMOD4
C                                                                           BELTRT
C     CONVERT TANGENT POINTS TO INERTIAL REFERENCE AND STORE.               BELTRT
C                                                                           BELTRT
   11 CALL DOT31 (D(1,1,I),APA,TPTS(1,M))                                   BELTRT
      CALL DOT31 (D(1,1,I),APB,TPTS(4,M))                                   BELTRT
      DO  12  K=1,3                                                         BELTRT
      TPTS(K  ,M) = TPTS(K  ,M) + SEGLP(K,I)                                BELTRT
   12 TPTS(K+3,M) = TPTS(K+3,M) + SEGLP(K,I)                                BELTRT
      SDOT = 0.0                                                            BELTRT
      NCF = NTAB(NT+5)                                                      BELTRT
      IF  (NCF.NE.0)  GO TO 15                                              BELTRT
C                                                                           BELTRT
C     ZERO BELT FRICTION, COMPUTE STRAIN AND FORCE OF ENTIRE BELT.          BELTRT
C                                                                           BELTRT
      B1213 = BELT(12,M)+BELT(13,M)                                         BELTRT
      S = (TL-B1213)/B1213                                                  BELTRT
      SA = S                                                                BELTRT
      SB = S                                                                BELTRT
      IF  (S.LT.0.0)  S = 0.0                                               BELTRT
      CALL FRCDFL (S,SDOT,NT,1,FA,ELOSS)                                    BELTRT
      FB = FA                                                               BELTRT
      GO TO 17                                                              BELTRT
C                                                                           BELTRT
C     FULL BELT FRICTION, COMPUTE STRAIN AND FORCE OF EACH PART OF BELT.BELTRT
C                                                                           BELTRT
   15 IF  (TL.GT.BELT(10,M))  GO TO 16                                      BELTRT
      FA = 0.0                                                              BELTRT
      FB = 0.0                                                              BELTRT
      SA = (TL-BELT(10,M))/BELT(10,M)                                       BELTRT
      SB = SA                                                               BELTRT
```

```
      BELT(12,M) = TLA                                                  BELTRT
      BELT(13,M) = TLB                                                  BELTRT
      GO TO 17                                                          BELTRT
   16 S = (TLA-BELT(12,M))/BELT(12,M)                                   BELTRT
      SA = S                                                            BELTRT
      IF  (S.LT.0.0)  S = 0.0                                           BELTRT
      CALL FRCDFL (S,SDOT,NT,1,FA,ELOSS)                                BELTRT
      S = (TLB-BELT(13,M))/BELT(13,M)                                   BELTRT
      SB = S                                                            BELTRT
      IF  (S.LT.0.0)  S = 0.0                                           BELTRT
      CALL FRCDFL (S,SDOT,NT+6,1,FB,ELOSS)                              BELTRT
      BELT(10,M) = 0.0                                                  BELTRT
   17 BSF(1,NBSF) = SA                                                  BELTRT
      BSF(2,NBSF) = FA                                                  BELTRT
      BSF(3,NBSF) = SB                                                  BELTRT
      BSF(4,NBSF) = FB                                                  BELTRT
      IF  (FA+FB.LE.0.0)  GO TO 31                                      BELTRT
C                                                                       BELTRT
C     COMPUTE FORCE VECTORS.                                            BELTRT
C                                                                       BELTRT
      DO  20  K=1,3                                                     BELTRT
      UVA (K) = FA*UVA(K)                                               BELTRT
   20 UVB(K) = FB*UVB(K)                                                BELTRT
C                                                                       BELTRT
C     CONVERT FORCES TO INERTIAL REFERENCE AND ADD TO U1 ARRAY.         BELTRT
C                                                                       BELTRT
      CALL DOT31(D(1,1,I),UVA,TT )                                      BELTRT
      CALL DOT31(D(1,1,I),UVB,TTT)                                      BELTRT
      DO 30 K=1,3                                                       BELTRT
       U1(K,MA) = U1(K,MA) - TT(K)                                      JTF984
       U1(K,MB) = U1(K,MB) - TTT(K)                                     JTF984
   30 U1(K,I) = U1(K,I)+TTT(K) + TT(K)                                  JTF984
C                                                                       BELTRT
C     CONVERT TORQUES TO LOCAL REFERENCE AND ADD TO U2 ARRAY.           BELTRT
C                                                                       BELTRT
       CALL MAT31(D(1,1,MA),TT,ZA)                                      JTF984
       CALL MAT31(D(1,1,MB),TTT,ZB)                                     JTF984
       CALL CROSS(BELT(1,M),ZA,TA)                                      JTF984
       CALL CROSS(BELT(4,M),ZB,TB)                                      JTF984
      CALL CROSS(APA,UVA,TT)                                            BELTRT
      CALL CROSS(APB,UVB,TTT)                                           BELTRT
      DO 40 K=1,3                                                       BELTRT
       U2(K,MA) = U2(K,MA) - TA(K)                                      JTF984
       U2(K,MB) = U2(K,MB) - TB(K)                                      JTF984
   40 U2(K,I) = U2(K,I)+(TT(K)+TTT(K))                                  BELTRT
   31 CONTINUE                                                          BELTRT
      CALL ELTIME(2,22)                                                 BELTRT
      RETURN                                                            BELTRT
      END                                                               BELTRT
```

```
      SUBROUTINE BGG(A,ZA,DA,BFA,VA,WA,                                 BGG
     *               B,ZB,DB,BFB,VB,WB,                                 BGG
     *               VSCS,IFULL,TV,FRA,TORQ,TQB,VOL)                    BGG
C                                                  REV IV   07/23/86TWOPI
C     COMPUTES THE VOLUME OF INTERSECTION OF AN ELLIPSOIDAL AIRBAG      BGG
C     WITH AN ELLIPSOIDAL BODY SEGMENT OR REACTION PANEL.               BGG
C     ALSO COMPUTES THE FORCE PER UNIT PRESSURE AND TORQUE PER UNIT     BGG
C     PRESSURE ON BOTH THE BAG AND THE INTERSECTING OBJECT.             BGG
C                                                                       BGG
C     ARGUMENTS:                                                        BGG
C      AIRBAG INPUTS : A(3,3) - ELLIPSOID MATRIX                        BGG
C                      ZA(3)  - C.G.                                    BGG
C                      DA(3,3)- DIRECTION COSINE MATRIX                 BGG
C                      BFA(3) - OFFSET                                  BGG
C                      VA(3)  - CG VELOCITY(INERTIAL REF.)              BGG
C                      WA(3)  - ANGULAR VELOCITY (LOCAL REF.)           BGG
C                                                                       BGG
C      CONTACT SURFACE B(3,3) - ELLIPSOID MATRIX                        BGG
C                      ZB(3)  - C.G.                                    BGG
C                      DB(3,3)- DIRECTION COSINE MATRIX                 BGG
C                      BFB(3) - OFFSET                                  BGG
C                      VB(3)  - CG VELOCITY (INERTIAL REF.)             BGG
C                      WB(3)  - ANGULAR VELOCITY (LOCAL REF.)           BGG
C                      VSCS   - COEFFICIENT OF SLIDING FRICTION         BGG
C                      IFULL  - IF ZERO, COMPUTE VOL ONLY.              BGG
C                      TV(3)  - MEMORY FOR SUBROUTINES INTERS & EDEPTH. BGG
C                                                                       BGG
C            OUTPUT  : FRA(3) - FORCE ON BAG                            BGG
C                      TORQ(3)- TORQUE ON BAG                           BGG
C                      TOB(3) - TORQUE ON CONTACT SURFACE               BGG
C                      VOL    - VOLUME OF INTERSECTION                  BGG
      IMPLICIT REAL*8 (A-H,O-Z)                                         BGG
      DIMENSION A(3,3),ZA(3),DA(3,3),BFA(3),VA(3),WA(3),B(3,3),ZB(3),   BGG
     *          DB(3,3),BFB(3),VB(3),WB(3),FRA(3),TORQ(3),TQB(3),TV(3)  BGG
      COMMON/CNSNTS/ PI,RADIAN,G,THIRD,EPS(24),                         BGG
     *               UNITL,UNITM,UNITT,GRAVTY(3),TWOPI                  TWOPI
      COMMON/TEMPVS/ DUMMY(200),DAB(3,3),BA(3,4),TEMP(3,3),Y(3),CPA(3), BGG
     *               CPB(3),PLANE(4,3),FORCE(3),CBB(3),VLM(3),FRB(3),   BGG
     *        YFA(3),YFB(3),ZBB(3),T1(3),T2(3),T3(3),T4(3),T5(3),T6(3)  BGG
C     NOTE: DUMMY IS USED BY SUBROUTINES AIRBAG AND AIRBGG.             BGG
C                                                                       BGG
C     INITIALIZATION                                                    BGG
C                                                                       BGG
      S3TEST = 10.0                                                     BGG
      VOL=0.                                                            BGG
      DO 5 I=1,3                                                        BGG
      FRA(I)  = 0.0                                                     BGG
      TORQ(I) = 0.0                                                     BGG
      TQB (I) = 0.0                                                     BGG
      BA(I,4)=-BFA(I)                                                   BGG
```

```
      DO 5 J=1,3                                                        BGG
      BA(I,4)=BA(I,4)+DA(I,J)*(ZB(J)-ZA(J))                             BGG
      DAB(I,J)=0.                                                       BGG
      DO 5 K=1,3                                                        BGG
    5 DAB(I,J)=DAB(I,J)+DA(I,K)*DB(J,K)                                 BGG
C                                                                       BGG
C     COMPUTE DISTANCE BETWEEN ELLIPSOID CENTERS AND                    BGG
C     CONVERT ELLIPSOID MATRIX OF OBJECT TO AIRBAG REFERENCE.           BGG
C                                                                       BGG
      DO 10 I=1,3                                                       BGG
      DO 10 J=1,3                                                       BGG
      TEMP(I,J) = 0.0                                                   BGG
      BA(I,4)=BA(I,4)+DAB(I,J)*BFB(J)                                   BGG
      DO 10 K=1,3                                                       BGG
   10 TEMP(I,J) = TEMP(I,J) + B(I,K)*DAB(J,K)                           BGG
      CALL MAT33(DAB,TEMP,BA)                                           BGG
C                                                                       BGG
C     CHECK FOR INTERSECTION AND DETERMINE POINTS OF MAXIMUM PENETRATIONBGG
C                                                                       BGG
      TB = 1.0                                                          BGG
      CALL INTERS (A,BA,BA(1,4),TB,Y,TV(1),T1)                          BGG
      IF (TB.GT.1.0) RETURN                                             BGG
      CALL EDEPTH (A,BA,BA(1,4),TB,Y,CPA,CPB,TV(2),TV(3))               BGG
C                                                                       BGG
C     SET UP ORTHOGONAL SYSTEM USING VECTOR BETWEEN POINTS              BGG
C     OF MAXIMUM PENETRATION AS ONE AXIS.                               BGG
C                                                                       BGG
      P = 0.                                                            BGG
      DO 20  I=1,3                                                      BGG
      PLANE(I,3) = CPA(I)-CPB(I)                                        BGG
   20 P = PLANE(I,3)**2+P                                               BGG
      IF  (P.LT.EPS(6))  GO TO 99                                       BGG
      PP = DSQRT(P)                                                     BGG
      DO 25  I=1,3                                                      BGG
   25 TEMP(I,1) = PLANE(I,3)/PP                                         BGG
      CALL ORTHO(PLANE,TEMP,4)                                          BGG
C                                                                       BGG
C     DEFINE PLANES AT MAXIMUM PENETRATION POINTS.                      BGG
C                                                                       BGG
      DO 40  I=1,3                                                      BGG
      PLANE(4,I) = 0.0                                                  BGG
      DO 40  J=1,3                                                      BGG
   40 PLANE(4,I) = PLANE(4,I)+PLANE(J,I)*CPB(J)                         BGG
      DO 45 K=1,3                                                       BGG
   45 CBB(K)=CPB(K)-BA(K,4)                                             BGG
C                                                                       BGG
C     ESTIMATES OF VOLUME AND AREA BASED ON RADII OF CURVATURE          BGG
C         AND PENETRATION.                                              BGG
C                                                                       BGG
      IP=2                                                              BGG
```

```
      AREA=PI                                                           BGG
      DO 70 L=1,2                                                       BGG
      RA=RCRT(A,PLANE,CPA,L)                                            BGG
      RB=RCRT(BA,PLANE,CBB,L)                                           BGG
      IF(PP.GT.RA)RA=PP                                                 BGG
      R=(RA-RB)*.5                                                      BGG
      RC=(RA+RB)*.5                                                     BGG
      VP=PP/(RA+RB)                                                     BGG
      VD=VP                                                             BGG
      ALP=RC*DSQRT(VP*(2.-VP))                                          BGG
      IF(R.GE.0.)GO TO 60                                               BGG
      AB=RA+RB-PP                                                       BGG
      BET=(RA**2-RB**2+AB**2)*.5/AB                                     BGG
      ALP=DSQRT(RA**2-BET**2)                                           BGG
      R=0.                                                              BGG
      VD=1.-BET/RA                                                      BGG
      VP=(PP+BET-RA)/RB                                                 BGG
   60 VLM(L)=RB*(RB*VP)**2*(1.-VP/3.)+RA*(RA*VD)**2*(1.-VD/3.)          BGG
      IF(R.GT.0.)VLM(L)=VLM(L)-ALP*R*R*(PI-2.*(DASIN(1.-VP)+            BGG
     *  (1.-VP)*ALP/RC))                                                BGG
      VLM(L)=VLM(L)*PI                                                  BGG
      AREA=AREA*ALP                                                     BGG
   70 IP=1                                                              BGG
      VOL=(VLM(1)+VLM(2))*.5                                            BGG
      IF  (IFULL.EQ.0)  GO TO 99                                        BGG
C                                                                       BGG
C     SET UP FORCE VECTOR ALONG LINE OF MAXIMUM PENETRATION.            BGG
C                                                                       BGG
      CALL DOT31(DAB,CBB,ZBB)                                           BGG
      DO 76 K=1,3                                                       BGG
      YFA(K)=CPB(K)+BFA(K)                                              BGG
      YFB(K)=ZBB(K)+BFB(K)                                              BGG
      FORCE(K) = -AREA*PLANE(K,3)                                       BGG
   76 T1(K) = VA(K)-VB(K)                                               BGG
C                                                                       BGG
C     COMPUTE ANGULAR VELOCITY COMPONENTS,RELATIVE VELOCITY, COMPONENTS BGG
C     OF RELATIVE VELOCITY ALONG MAX PENETRATION LINE AND MAGNITUDE OF  BGG
C     FORCE.                                                            BGG
C                                                                       BGG
      CALL MAT31(DA,T1,T2)                                              BGG
      CALL CROSS(WA,YFA,T1)                                             BGG
      CALL CROSS(WB,YFB,T3)                                             BGG
      CALL MAT31(DAB,T3,T4)                                             BGG
      FM = 0.0                                                          BGG
      SUM = 0.0                                                         BGG
      DO 77 K=1,3                                                       BGG
      T5(K) = T2(K)+T1(K)-T4(K)                                         BGG
      SUM = SUM+T5(K)*PLANE(K,3)                                        BGG
   77 FM = FM+FORCE(K)**2                                               BGG
C                                                                       BGG
```

```
C     COMPUTE COMPONENTS OF RELATIVE VELOCITY IN TANGENT PLANE,          BGG
C     FRICTION FORCE AND TOTAL FORCE VECTOR.                             BGG
C                                                                        BGG
      S3 = 0.0                                                           BGG
      DO 78 K=1,3                                                        BGG
      T6(K) = T5(K)-SUM*PLANE(K,3)                                       BGG
   78 S3 = S3+T6(K)**2                                                   BGG
      SQ3 = DSQRT(S3)                                                    BGG
      IF (SQ3.LT.S3TEST) SQ3=S3TEST/(2.0-SQ3/S3TEST)                     BGG
      FF = VSCS*DSQRT(FM)/SQ3                                            BGG
      DO 79 K=1,3                                                        BGG
   79 FORCE(K) = FORCE(K)-FF*T6(K)                                       BGG
C                                                                        BGG
C     COMPUTE FRB: FORCE ON REACTION SURFACE IN ITS LOCAL REFERENCE.     BGG
C            TORQ: TORQUE ON AIRBAG IN AIRBAG REFERENCE.                 BGG
C             TQB: TORQUE ON REACTION SURFACE IN ITS LOCAL REFERENCE.    BGG
C             FRA: FORCE ON AIRBAG IN INERTIAL REFERENCE.                BGG
C                                                                        BGG
      CALL DOT31(DAB,FORCE,FRB)                                          BGG
      CALL CROSS(YFA,FORCE,TORQ)                                         BGG
      CALL CROSS(FRB,YFB,TQB)                                            BGG
      CALL DOT31(DA,FORCE,FRA)                                           BGG
   99 RETURN                                                             BGG
      END                                                                BGG
```

```
      SUBROUTINE BINPUT                                                 BINPUT
C                                                 REV IV    07/24/86SLIP
C     READS THE INPUT CARDS THAT CONTAINS THE PHYSICAL DIMENSIONS AND   BINPUT
C     CHARACTERISTICS OF THE CRASH VICTIM'S BODY SEGMENTS AND JOINTS.   BINPUT
C                                                                       BINPUT
      IMPLICIT REAL*8(A-H,O-Z)                                          BINPUT
      COMMON/CONTRL/ TIME,NSEG,NJNT,NPL,NBLT,NBAG,NVEH,NGRND,           BINPUT
     *               NS,NQ,NSD,NFLX,NHRNSS,NWINDF,NJNTF,NPRT(36),NPG    PAGE
      COMMON/DESCRP/ PHI(3,30),W(30),RW(30),SR(4,60),HA(3,60),HB(3,60), SLIP
     *               RPHI(3,30),HT(3,3,60),SPRING(5,90),VISC(7,90),     BINPUT
     *               JNT(30),IPIN(30),ISING(30),IGLOB(30),JOINTF(30)    BINPUT
      COMMON/CNTSRF/ PL(24,30),BELT(20,8),TPTS(6,8),BD(24,40)           EDGE
      COMMON/TITLES/ DATE(3),COMENT(40),VPSTTL(20),BDYTTL(5),           BINPUT
     *               BLTTTL(5,8),PLTTL(5,30),BAGTTL(5,6),SEG(30),       BINPUT
     *               JOINT(30),CGS(30),JS(30)                           BINPUT
      REAL DATE,COMENT,VPSTTL,BDYTTL,BLTTTL,PLTTL,BAGTTL,SEG,JOINT      BINPUT
      LOGICAL*1 CGS,JS                                                  BINPUT
      COMMON/INTEST/ SGTEST(3,4,30),XTEST(3,120),SEGT(120),REGT(120)    BINPUT
      REAL    SEGT                                                      BINPUT
      COMMON/FLXBLE/ HF(4,12,8),B42(3,3,24),V4(3,8),NFLEX(3,8)          BINPUT
      COMMON/CEULER/ IEULER(30),HIR(3,3,90),ANG(3,30),ANGD(3,30),       JDRIFT
     *               FE(3,30),TQE(3,30),CONST(5,30)                     JDRIFT
      COMMON/CNSNTS/ PI,RADIAN,G,THIRD,EPS(24),                         BINPUT
     *               UNITL,UNITM,UNITT,GRAVTY(3),TWOPI                  TWOPI
      COMMON/RSAVE/  XSG(3,20,3),DPMI(3,3,30),LPMI(30),                 ATBIII
     *               NSG(9),MSG(20,9),MCG,MCGIN(24,5),KREF(20,9)        TTHKREF
      LOGICAL*1 EULER,SLIP                                              SLIP
      COMMON/TEMPVS/ YPR1(3,30),YPR2(3,30),YPR3(3,30),YPRPMI(3,30),     BINPUT
     *               T1(6),TMP1(3,3),TMP2(3,3),KNT(30),IDYPR(6,30),     SLIP
     *               EULER(30)                                          SLIP
      DATA MXNSEG/30/,MXNJNT/30/,MNFLX/8/                               MISC
      CALL ELTIME(1, 2)                                                 BINPUT
      IDYPRT = 0                                                        TGMOD5
C                                                                       BINPUT
C     INPUT CARD B.1                                                    BINPUT
C                                                                       BINPUT
      READ (5,11) NSEG,NJNT,BDYTTL                                      BINPUT
   11 FORMAT (2I6,8X,5A4)                                               BINPUT
      IF (NSEG.GT.MXNSEG) STOP 77                                       MISC
      IF (NJNT.GT.MXNJNT) STOP 78                                       MISC
C                                                                       BINPUT
C     INPUT CARDS  B.2.I  FOR EACH SEGMENT.                             BINPUT
C                                                                       BINPUT
      DO  12  I=1,NSEG                                                  BINPUT
      READ (5,13) SEG(I),CGS(I),W(I),(PHI(J,I),J=1,3),                  BINPUT
     *            (BD(J,I),J=1,6),LPMI(I)                               BINPUT
   13 FORMAT(A4,1X,A1,10F6.0,I4)                                        BINPUT
      DO  81  J=1,3                                                     BINPUT
      IDYPR(J,I) = 4-J                                                  BINPUT
   81 YPRPMI(J,I) = 0.0                                                 BINPUT
```

```
      IF  (LPMI(I).EQ.0)  GO TO 12                                      BINPUT
      READ (5,82) (YPRPMI(J,I),J=1,3)                                   BINPUT
   82 FORMAT(12X,3F6.0)                                                 BINPUT
   12 CALL DRCYPR (DPMI(1,1,I),YPRPMI(1,I),IDYPR(1,I))                  BINPUT
C                                                                       BINPUT
C     INPUT CARDS  B.3.J  FOR EACH JOINT.                               BINPUT
C                                                                       BINPUT
      NFLX = 0                                                          BINPUT
      IF (NJNT.EQ.0) GO TO 27                                           BINPUT
      SLIP = .FALSE.                                                    SLIP
      DO  14  J=1,NJNT                                                  BINPUT
      READ (5,15) JOINT(J),JS(J),JNT(J),IPIN(J),(SR(I,2*J-1),I=1,3),    BINPUT
     *            (SR(I,2*J),I=1,3),IEULER(J),CONST(1,J),CONST(2,J),    SLIP
     *            (YPR1(I,J),I=1,3),(YPR2(I,J),I=1,3),                  SLIP
     *            (YPR3(I,J),I=1,3),(IDYPR(I,J),I=1,6)                  BINPUT
      ID1 = IDYPR(1,J)                                                  BINPUT
      ID4 = IDYPR(4,J)                                                  BINPUT
      EULER(J) = .FALSE.                                                SLIP
      IF (IPIN(J).EQ.4) EULER(J) = .TRUE.                               SLIP
      IF (IEULER(J).EQ.0.AND.IPIN(J).LE.-4) EULER(J) = .TRUE.           SLIP
      IF (.NOT.EULER(J).AND.IABS(IPIN(J)).GE.5) SLIP = .TRUE.           SLIP
      IF(ID1.NE.0.OR.ID4.NE.0) IDYPRT = 1                               TGMOD5
      DO 479 II=1,6                                                     TGMOD5
  479 IF(IABS(IDYPR(II,J)).GT.3) STOP 101                               TGMOD5
      DO  14  I=1,3                                                     BINPUT
      IF  (ID1.EQ.0)  IDYPR(I  ,J) = 4-I                                BINPUT
   14 IF  (ID4.EQ.0)  IDYPR(I+3,J) = 4-I                                BINPUT
   15 FORMAT(A4,1X,A1,2I4,6F6.0,I4,2F6.0/14X,9F6.0,6I2)                 SLIP
C                                                                       BINPUT
C     COMPUTE NFLX AND NFLEX ARRAY FROM NEGATIVE VALUES OF JNT(J).      BINPUT
C     NFLX WILL BE NUMBER OF CONSTRAINT TORQUES FOR FLEXIBLE SEGMENTS.  BINPUT
C     NFLEX(1, ) REFERENCE SEGMENT (LOWEST NUMBERED SEGMENT OF CHAIN)   BINPUT
C     NFLEX(2, ) INTERIOR SEGMENT NUMBERS                               BINPUT
C     NFLEX(3, ) TERMINATING SEGMENT (HIGHEST NUMBERED SEGMENT IN CHAIN)BINPUT
C     VALUES OF NFLEX NEED NOT BE SEQUENTIAL BUT MUST BE ORDERED.       BINPUT
C     FLEXIBLE SEGMENT MUST BE SIMPLE CHAIN, I.E., BRANCHING SEGMENTS   BINPUT
C     CANNOT BE ATTACHED TO INTERIOR SEGMENTS BUT MAY BE ATTACHED TO    BINPUT
C     REFERENCE OR TERMINATING SEGMENTS.                                BINPUT
C                                                                       BINPUT
      DO  16  J=1,NJNT                                                  BINPUT
   16 KNT(J) = JNT(J)                                                   BINPUT
      DO  22  J=1,NJNT                                                  BINPUT
      IF (KNT(J).GE.0) GO TO 22                                         BINPUT
      NFA = NFLX+1                                                      BINPUT
      IT = J+1                                                          BINPUT
      IF (IT.GT.NJNT) GO TO 18                                          BINPUT
      JP1 = J+1                                                         BINPUT
      DO  17  L=JP1,NJNT                                                BINPUT
      IF (IABS(KNT(L)).NE.IT) GO TO 17                                  BINPUT
      KL = KNT(L)                                                       BINPUT
```

```
      KNT(L) = 0                                                         BINPUT
      IF (KL.GT.0) GO TO 18                                              BINPUT
      NFLX = NFLX+1                                                      BINPUT
      NFLEX(1,NFLX) = IABS(KNT(J))                                       BINPUT
      NFLEX(2,NFLX) = IT                                                 BINPUT
      IT = L+1                                                           BINPUT
   17 CONTINUE                                                           BINPUT
   18 IF (NFLX.GE.NFA) GO TO 20                                          BINPUT
      WRITE (6,19)                                                       BINPUT
   19 FORMAT('0ERROR IN DEFINING FLEXIBLE SEGMENTS, ONLY ONE NEGATIVE JNBINPUT
     *T IN STRING. PROGRAM TERMINATED.')                                 BINPUT
      STOP 3                                                             BINPUT
   20 DO  21  K=NFA,NFLX                                                 BINPUT
   21 NFLEX(3,K) = IT                                                    BINPUT
   22 CONTINUE                                                           BINPUT
C                                                                        BINPUT
C     INPUT CARDS  B.4.J  FOR EACH JOINT.                                BINPUT
C                                                                        BINPUT
      DO  23  J=1,NJNT                                                   BINPUT
      READ (5,24) (SPRING(I,3*J-2),I=1,5),(SPRING(I,3*J-1),I=1,5)        BINPUT
   23 IF (EULER(J)) READ(5,24)(SPRING(I,3*J),I=1,5),(ANG(I,J),I=1,3)     SLIP
   24 FORMAT(2(4F6.0,F12.0))                                             BINPUT
C                                                                        BINPUT
C     INPUT CARDS  B.5.J  FOR EACH JOINT.                                BINPUT
C                                                                        BINPUT
      DO  25  J=1,NJNT                                                   BINPUT
      READ (5,26) (VISC(I,3*J-2),I=1,7)                                  BINPUT
      IF (.NOT.EULER(J)) GO TO 25                                        SLIP
      READ (5,26) (VISC(I,3*J-1),I=1,7)                                  BINPUT
      READ (5,26) (VISC(I,3*J  ),I=1,7)                                  BINPUT
   25 CONTINUE                                                           BINPUT
   26 FORMAT(5F6.0,18X,2F6.0)                                            BINPUT
C                                                                        BINPUT
C     INPUT CARDS  B.6.I  FOR EACH SEGMENT.                              BINPUT
C                                                                        BINPUT
   27 DO  28  I=1,NSEG                                                   BINPUT
   28 READ (5,29) ((SGTEST(J,K,I),J=1,3),K=1,4)                          BINPUT
   29 FORMAT(12F6.0)                                                     BINPUT
C                                                                        BINPUT
C     PRINT CARD B.1                                                     BINPUT
C                                                                        BINPUT
      WRITE (6,30) BDYTTL,NSEG,NJNT,NPG,UNITM,UNITT,UNITL,UNITL,         PAGE
     *             UNITL,UNITM                                           PAGE
      NPG=NPG+1                                                          PAGE
   30 FORMAT('1 CRASH VICTIM',5X,5A4,I5,' SEGMENTS',I5,' JOINTS',58X,    PAGE
     *   'PAGE',I5/120X,'CARD  B.1'/25X,'PRINCIPAL MOMENTS OF INERTIA',  PAGE
     *       14X,'SEGMENT CONTACT ELLIPSOID',28X,'CARDS B.2'/            BINPUT
     *       3X,'SEGMENT',6X,'WEIGHT',7X,'(',A4,'-',A4,'**2-',A4,')',    BINPUT
     *       11X,'SEMIAXES (',A4,')',12X,'CENTER (',A4,')',              BINPUT
     *       11X,'PRINCIPAL AXES (DEG)'/                                 BINPUT
```

```
     *        '  I  SYM PLOT   (',A4,')',7X,'X',8X,'Y',8X,'Z ',          BINPUT
     *        2(9X,'X',7X,'Y',7X,'Z'),8X,'YAW',5X,'PITCH',5X,'ROLL'/)    BINPUT
C                                                                        BINPUT
C     PRINT CARDS  B.2.I  FOR EACH SEGMENT.                              BINPUT
C                                                                        BINPUT
      DO  31  I=1,NSEG                                                   BINPUT
   31 WRITE (6,32) I,SEG(I),CGS(I),W(I),(PHI(J,I),J=1,3),                BINPUT
     *             (BD(J,I),J=1,6),(YPRPMI(J,I),J=1,3)                   BINPUT
   32 FORMAT(I3,1X,A4,2X,A1,F11.3,2X,3F9.4,2(2X,3F8.3),1X,3F9.2)         BINPUT
      IF (NJNT.EQ.0) GO TO 50                                            BUTLER1
C                                                                        BINPUT
C     PRINT CARDS  B.3.J  FOR EACH JOINT.                                BINPUT
C                                                                        BINPUT
      IF(IDYPRT.EQ.0) WRITE(6,33) UNITL,UNITL                            TGMOD5
      IF(IDYPRT.EQ.1) WRITE(6,733) UNITL,UNITL                           TGMOD5
   33 FORMAT(///120X,'CARDS B.3'/                                        BINPUT
     * 3X,'JOINT',15X,'LOCATION(',A4,') - SEG(JNT)',                     BINPUT
     *             3X,'LOCATION(',A4,') - SEG(J+1)',                     BINPUT
     *             2X, 'PRIN. AXIS(DEG) - SEG(JNT)',                     BINPUT
     *             2X, 'PRIN. AXIS(DEG) - SEG(J+1)'/                     BINPUT
     * '  J SYM PLOT JNT PIN', 2(6X,'X',8X,'Y',8X,'Z',3X),               BINPUT
     *             2(5X,'YAW',5X,'PITCH',5X,'ROLL',1X) /)                BINPUT
  733 FORMAT(///120X,'CARDS B.3'/                                        TGMOD5
     * 3X,'JOINT',15X,'LOCATION(',A4,') - SEG(JNT)',                     TGMOD5
     *             3X,'LOCATION(',A4,') - SEG(J+1)',                     TGMOD5
     *             2X, 'PRIN. AXIS(DEG) - SEG(JNT)',                     TGMOD5
     *             2X, 'PRIN. AXIS(DEG) - SEG(J+1)'/                     TGMOD5
     * '  J SYM PLOT JNT PIN', 2(6X,'X',8X,'Y',8X,'Z',3X),               TGMOD5
     *             'ID1  YAW ID2 PITCH ID3 ROLL ',                       TGMOD5
     *             'ID4  YAW ID5 PITCH ID6 ROLL ',/)                     TGMOD5
      DO  34  J=1,NJNT                                                   BINPUT
      IF(IDYPRT.EQ.0)                                                    TGMOD5
     *WRITE (6,35) J,JOINT(J),JS(J),JNT(J),IPIN(J),(SR(I,2*J-1),I=1,3), TGMOD5
     *             (SR(I,2*J),I=1,3),(YPR1(I,J),I=1,3),(YPR2(I,J),I=1,3)BINPUT
      IF(IDYPRT.EQ.1)                                                    TGMOD5
     *WRITE(6,735) J,JOINT(J),JS(J),JNT(J),IPIN(J),(SR(I,2*J-1),I=1,3), TGMOD5
     *             (SR(I,2*J),I=1,3),(IDYPR(I,J),YPR1(I,J),I=1,3),       TGMOD5
     *             (IDYPR(I+3,J),YPR2(I,J),I=1,3)                        TGMOD5
      IF (.NOT.EULER(J)) GO TO 34                                        SLIP
      IEULER(J) = 8                                                      BINPUT
      IF (IPIN(J).EQ.4) GO TO 34                                         BINPUT
      IEULER(J) = 11 + IPIN(J)                                           BINPUT
      IPIN(J) = -4                                                       BINPUT
   34 CONTINUE                                                           BINPUT
   35 FORMAT(I3,1X,A4,2X,A1,2X,2I3,2(1X,3F9.3),2(1X,3F9.2) )             BINPUT
  735 FORMAT(I3,1X,A4,2X,A1,2X,2I3,2(1X,3F9.3),2(1X,3(1X,I1,F7.2)))      TGMOD5
      IF (.NOT.SLIP) GO TO 89                                            SLIP
      WRITE (6,83) UNITM,UNITM                                           SLIP
   83 FORMAT(//'  UNLOCK CONDITIONS FOR SLIP JOINTS'/                    SLIP
     *        '      JOINT    TENSION    COMPRESSION'/                   SLIP
```

```
     *   14X,'(',A4,')',7X,'(',A4,')'/)                                     SLIP
      DO 85 J = 1,NJNT                                                      SLIP
      IF (EULER(J)) GO TO 85                                                SLIP
      IF (IABS(IPIN(J)).LT.5) GO TO 85                                      SLIP
      WRITE (6,84) J,CONST(1,J),CONST(2,J)                                  SLIP
   84 FORMAT(1X,I6,4X,F10.3,3X,F10.3)                                       SLIP
   85 CONTINUE                                                              SLIP
C                                                                           BINPUT
C     SET UP HT MATRIX FROM YPR1 & YPR2 INPUT.                              BINPUT
C     HA IS 3RD COLUMN & HB IS 2ND COLUMN OF HT.                            BINPUT
C     FOR A SLIP JOINT(IPIN=7),HB IS 3RD COLUMN OF HT.                      SLIP
C                                                                           BINPUT
89    IF  (NPRT(23).NE.0)  WRITE (6,36) NPG                                 SLIP
      IF  (NPRT(23).NE.0)  NPG=NPG+1                                        PAGE
   36 FORMAT('1 HT ARRAY AS COMPUTED FROM YPR1 & YPR2 INPUT.',77X,          PAGE
     *        'PAGE',I5)                                                    PAGE
      DO  38  J=1,NJNT                                                      BINPUT
      SR(4,2*J-1) = 0.0                                                     SLIP
      SR(4,2*J  ) = 0.0                                                     SLIP
      CALL DRCYPR (TMP1,YPR1(1,J),IDYPR(1,J))                               BINPUT
      CALL DRCYPR (TMP2,YPR2(1,J),IDYPR(4,J))                               BINPUT
      DO 37 I=1,3                                                           BINPUT
      ANGD(I,J) = 0.0                                                       BINPUT
      HA(I,2*J-1) = 0.0                                                     BINPUT
      HA(I,2*J  ) = 0.0                                                     BINPUT
      K = 2                                                                 SLIP
      IF (IABS(IPIN(J)).EQ.7) K = 3                                         SLIP
      HB(I,2*J-1) = TMP1(K,I)                                               SLIP
      HB(I,2*J  ) = TMP2(K,I)                                               SLIP
      DO 77 K=1,3                                                           SLIP
      HT(I,K,2*J-1) = TMP1(K,I)                                             SLIP
   77 HT(I,K,2*J  ) = TMP2(K,I)                                             SLIP
      IF (.NOT.EULER(J)) GO TO 37                                           SLIP
      CONST(I,J) = YPR3(I,J)*RADIAN                                         SLIP
      ANG(I,J) = ANG(I,J)*RADIAN - CONST(I,J)                               SLIP
   37 CONTINUE                                                              SLIP
   38 IF  (NPRT(23).NE.0)  WRITE (6,39) J,JOINT(J),                         BINPUT
     *    ((HT(I,K,2*J-1),K=1,3),(HT(I,K,2*J),K=1,3),I=1,3)                 BINPUT
   39 FORMAT('0',I4,2X,A4,3X,3F12.6,3X,3F12.6/(14X,3F12.6,3X,3F12.6))       BINPUT
C                                                                           BINPUT
C     PRINT CARDS  B.4.J  FOR EACH JOINT.                                   BINPUT
C                                                                           BINPUT
      WRITE (6,41) NPG,UNITL,UNITM,UNITL,UNITM                              PAGE
      NPG=NPG+1                                                             PAGE
   41 FORMAT('1 JOINT TORQUE CHARACTERISTICS',93X,                          PAGE
     *        'PAGE',I5/120X,'CARDS B.4'/                                   PAGE
     *23X,'FLEXURAL SPRING CHARACTERISTICS',28X,'TORSIONAL SPRING' ,        BINPUT
     *' CHARACTERISTICS'//                                                  BINPUT
     *15X,'SPRING COEF. (',2A4,'/DEG**J)',6X,'ENERGY     JOINT',            BINPUT
     * 7X,'SPRING COEF. (',2A4,'/DEG**J)',6X,'ENERGY     JOINT'/     '     JBINPUT
```

```
     *OINT ',2(8X,'LINEAR    QUADRATIC     CUBIC    DISSIPATION  STOP ')BINPUT
     */8X,2(8X,'(J=1)',7X,'(J=2)',7X,'(J=3)',7X,'COEF.      (DEG)')/)   BINPUT
      DO  42  J=1,NJNT                                                 BINPUT
      J1 = 3*J-2                                                       BINPUT
      J2 = 3*J-1                                                       BINPUT
      J3 = 3*J                                                         BINPUT
      WRITE (6,43) J,JOINT(J),((SPRING(I,JJ),I=1,5),JJ=J1,J2)          BINPUT
   42 IF (EULER(J)) WRITE (6,44) (SPRING(I,J3),I=1,5)                  SLIP
   43 FORMAT(I3,1X,A4,2(3X,3F12.3,2F10.3))                             BINPUT
   44 FORMAT(11X,3F12.3,2F10.3)                                        BINPUT
C                                                                      BINPUT
C     PRINT CARDS B.5.J  FOR EACH JOINT.                               BINPUT
C                                                                      BINPUT
      WRITE (6,46) (UNITL,UNITM,UNITT,I=1,2),(UNITL,UNITM,I=1,2),UNITT BINPUT
   46 FORMAT(///120X,'CARDS B.5'/                                      BINPUT
     *38X,'JOINT VISCOUS CHARACTERISTICS AND LOCK-UNLOCK CONDITIONS'// BINPUT
     *14X,'VISCOUS',9X,'COULOMB',7X,'FULL FRICTION',5X,'MAX TORQUE FOR',BINPUT
     *4X,'MIN TORQUE FOR',4X,'MIN. ANG. VELOCITY',6X,'IMPULSE'/        BINPUT
     *2X,'JOINT',5X,'COEFFICIENT',4X,'FRICTION COEF. ANGULAR VELOCITY', BINPUT
     *4X,'A LOCKED JOINT',4X,'UNLOCKED JOINT',4X,'FOR UNLOCKED JOINT', BINPUT
     *4X,'RESTITUTION'/                                                BINPUT
     *8X,'(',3A4,'/DEG)  (',2A4,')',6X,'(DEG/',A4,')',10X,'(',2A4,')', BINPUT
     *8X,'(',2A4,')',10X,'(RAD/',A4,')',8X,'COEFFICIENT'/ )            BINPUT
      DO  47  J=1,NJNT                                                 BINPUT
      J1 = 3*J-2                                                       BINPUT
      J2 = 3*J-1                                                       BINPUT
      J3 = 3*J                                                         BINPUT
      WRITE (6,48) J,JOINT(J),(VISC(I,J1),I=1,7)                       BINPUT
   47 IF (EULER(J)) WRITE (6,49) ((VISC(I,JJ),I=1,7),JJ=J2,J3)         SLIP
   48 FORMAT(I3,1X,A4,F13.3,2F15.2,F22.2,F18.2,F20.2,F17.3)            BINPUT
   49 FORMAT(       8X,F13.3,2F15.2,F22.2,F18.2,F20.2,F17.3)           BINPUT
C                                                                      BINPUT
C     PRINT CARDS  B.6.I  FOR EACH SEGMENT.                            BINPUT
C                                                                      BINPUT
   50 WRITE (6,51) NPG,(UNITT,UNITL,UNITT,I=1,2)                       PAGE
      NPG=NPG+1                                                        PAGE
   51 FORMAT('1',122X,'PAGE',I5/20X,                                   PAGE
     *     'SEGMENT INTEGRATION CONVERGENCE TEST INPUT',58X,'CARDS B.6'//PAGE
     *      17X,'ANGULAR VELOCITIES',  11X,'LINEAR VELOCITIES',        BINPUT
     *      10X,'ANGULAR ACCELERATIONS',9X,'LINEAR ACCELERATIONS'/     BINPUT
     *     21X,'(RAD/',A4,')', 18X,'(',A4,'/',A4,')',                  BINPUT
     *     17X,'(RAD/',A4,'**2)', 16X,'(',A4,'/',A4,'**2)'/            BINPUT
     *     ' SEGMENT', 4('       MAG.     ABS.      REL.')    /         BINPUT
     *     ' NO. SYM', 4('       TEST     ERROR     ERROR')   /)        BINPUT
      DO 52 I=1,NSEG                                                   BINPUT
   52 WRITE (6,53) I,SEG(I),((SGTEST(J,K,I),J=1,3),K=1,4)              BINPUT
   53 FORMAT(I3,1X,A4,4(F11.3,F9.3,F9.4) )                             BINPUT
      IF (NFLX.EQ.0) GO TO 62                                          BINPUT
C                                                                      BINPUT
C     INPUT AND PRINT CARDS B.7                                        BINPUT
```

```
C     CARD B.7.A  NFX: NO. OF INTERIOR SEGMENTS OF FLEXIBLE ELEMENTS.   BINPUT
C                 KNT(J),J=1,NFX: THE SEGMENT NUMBERS.                  BINPUT
C                                                                       BINPUT
      READ (5,54) NFX,(KNT(J),J=1,NFX)                                  BINPUT
   54 FORMAT(18I4)                                                      BINPUT
      IF (NFX.NE.NFLX) WRITE (6,55) NFX,NFLX                            BINPUT
   55 FORMAT('OINPUT ERROR ON CARD B.7.A, NFX =',I4, ' BUT NFLX =',I4/  BINPUT
     *        ' AS COMPUTED FROM CARDS B.3. PROGRAM TERMINATED.')       BINPUT
      IF  (NFX.NE.NFLX)  STOP 4                                         BINPUT
      WRITE (6,56) NPG                                                  PAGE
      NPG=NPG+1                                                         PAGE
   56 FORMAT('1',122X,'PAGE',I5/121X,'CARDS B.7')                       PAGE
      DO 60 JJ=1,NFX                                                    BINPUT
      DO  57  K=1,NFLX                                                  BINPUT
      IF (KNT(JJ).EQ.NFLEX(2,K)) GO TO 59                               BINPUT
   57 CONTINUE                                                          BINPUT
      WRITE (6,58) KNT(JJ)                                              BINPUT
   58 FORMAT('OINPUT ERROR ON CARD B.7.J, SEGMENT NO.',I4,' IS NOT AN INBINPUT
     *TERIOR SEGMENT OF A FLEXIBLE ELEMENT FROM DATA ON CARDS B.3.'/    BINPUT
     *        ' PROGRAM TERMINATED.')                                   BINPUT
      STOP 5                                                            BINPUT
   59 IF(NFLX.GT.MNFLX) STOP 99                                         TGMOD5
C                                                                       BINPUT
C     CARDS B.7.J  HF ARRAY FOR SEGMENT KNT(JJ)                         BINPUT
C                                                                       BINPUT
      READ (5,29) ((HF(I,J,K),J=1,12),I=1,4)                            TGMOD5
      DO 737 LL=1,3                                                     TGMOD5
      L = (LL-1)*4                                                      TGMOD5
      DO 737 I=1,4                                                      TGMOD5
      DO 737 J=1,4                                                      TGMOD5
  737 IF(HF(I,J+L,K).NE.HF(J,I+L,K)) STOP 100                           TGMOD5
   60 WRITE (6,61) KNT(JJ),K,(NFLEX(I,K),I=1,3),                        BINPUT
     *             ((HF(I,J,K),J=1,12),I=1,4)                           BINPUT
   61 FORMAT('0 HF ARRAY FOR INTERIOR SEGMENT NO.',I4,20X,              BINPUT
     *       '(NFLEX(I,',I1,'),I=1,3) =',3I6//                          BINPUT
     *       (3X,4F10.4,3X,4F10.4,3X,4F10.4) )                          BINPUT
   62 IF (NJNT.EQ.0) GO TO 65                                           BINPUT
C                                                                       BINPUT
C     CHANGE SPRING AND VISC FROM DEG TO RAD                            BINPUT
C                                                                       BINPUT
      DO  64  I=1,NJNT                                                  BINPUT
      J1 = 3*I-2                                                        BINPUT
      J2 = 3*I-1                                                        BINPUT
      IF (EULER(I)) J2= 3*I                                             SLIP
      DO 63 J=J1,J2                                                     BINPUT
      SPRING(1,J) = SPRING(1,J)/RADIAN                                  BINPUT
      SPRING(2,J) = SPRING(2,J)/RADIAN**2                               BINPUT
      SPRING(3,J) = SPRING(3,J)/RADIAN**3                               BINPUT
      SPRING(5,J) = SPRING(5,J)*RADIAN                                  BINPUT
   63 CONTINUE                                                          BINPUT
```

```
      IF (.NOT.EULER(I)) J2 = J1                                        SLIP
      DO 64 J=J1,J2                                                     BINPUT
      VISC  (1,J) = VISC  (1,J)/RADIAN                                  BINPUT
   64 VISC  (3,J) = VISC  (3,J)*RADIAN                                  BINPUT
C                                                                       BINPUT
C     W ARRAY HAS BEEN SUPPLIED IN LBS. SET UP RECIPROCAL MASS (RW)     BINPUT
C     AND MOMENT OF INERTIA (RPHI) ARRAYS. HOWEVER, IF W OR ANY ELEMENT BINPUT
C     OF PHI IS ZERO, SEGMENT WILL BE CONSIDERED SINGULAR (ISING=1) AND BINPUT
C     ALL RECIPROCALS WILL BE ZERO SO AS TO NULLIFY COMPUTATIONS IN THE BINPUT
C     DAUX ROUTINES. NS IS THE NUMBER OF SINGULAR SEGMENTS.             BINPUT
C                                                                       BINPUT
   65 NS = 0                                                            BINPUT
      DO  68  I=1,NSEG                                                  BINPUT
      ISING(I) = 0                                                      BINPUT
      RW(I) = 0.0                                                       BINPUT
      IF (W(I).EQ.0.0) ISING(I) = 1                                     BINPUT
      DO 66 K=1,3                                                       BINPUT
      IF (PHI(K,I).EQ.0.0) ISING(I) = 1                                 BINPUT
   66 RPHI(K,I) = 0.0                                                   BINPUT
      IF (ISING(I).EQ.1) NS = NS+1                                      BINPUT
      IF (ISING(I).EQ.1) GO TO 68                                       BINPUT
      RW(I) = G/W(I)                                                    BINPUT
      DO 67 K=1,3                                                       BINPUT
   67 RPHI(K,I) = 1.0/PHI(K,I)                                          BINPUT
   68 CONTINUE                                                          BINPUT
C                                                                       BINPUT
C     SET UP ELLIPSOID MATRIX AND INVERSE (ASSUME YAW,PITCH,ROLL = 0)   BINPUT
C     FOR 1ST NSEG ELLIPSOIDS IN BD(7-15) AND BD(16-24).                BINPUT
C                                                                       BINPUT
      DO  71  J=1,NSEG                                                  BINPUT
      DO 70 I=7,24                                                      BINPUT
   70 BD(I,J) = 0.0                                                     BINPUT
      DO 71 I=1,3                                                       BINPUT
      BD(4*I+3,J) = 1.0/BD(I,J)**2                                      BINPUT
   71 BD(4*I+12,J) = BD(I,J)**2                                         BINPUT
      RETURN                                                            BINPUT
      END                                                               BINPUT
```

```
      SUBROUTINE BLKDTA                                                 BLKDTA
C                                                   REV IV    07/23/86TWOPI
C     THIS SUBROUTINE REPLACES THE BLOCK DATA SUBPROGRAM OF PREVIOUS    BLKDTA
C     VERSIONS OF CVS-III TO INITIALIZE COMMON/CNSNTS/ IN A MANNER      BLKDTA
C     THAT IS INDEPENDENT OF THE COMPUTER SYSTEM BEING UTILIZED.        BLKDTA
C                                                                       BLKDTA
      IMPLICIT REAL*8 (A-H,O-Z)                                         BLKDTA
      COMMON/CNSNTS/ PI,RADIAN,G,THIRD,EPS(24),                         BLKDTA
     *               UNITL,UNITM,UNITT,GRAVTY(3),TWOPI                  TWOPI
      COMMON/TEMPVS/ ZERO,ONE,THREE,TEN,ONE80                           BLKDTA
      DATA  UM/8H LBS    / , UT/8H SEC    / , UL/8H IN     /            BLKDTA
      ZERO  = 0.0                                                       BLKDTA
      ONE   = 1.0                                                       BLKDTA
      UNITM = UM                                                        BLKDTA
      UNITT = UT                                                        BLKDTA
      UNITL = UL                                                        BLKDTA
      G     = 386.088D0                                                 BLKDTA
      GRAVTY(1) = ZERO                                                  BLKDTA
      GRAVTY(2) = ZERO                                                  BLKDTA
      GRAVTY(3) = G                                                     BLKDTA
      THREE = 3.0                                                       BLKDTA
      TEN   = 10.0                                                      BLKDTA
      ONE80 = 180.0                                                     BLKDTA
      PI = DATAN2(ZERO,-ONE)                                            BLKDTA
      TWOPI = 2.0*PI                                                    TWOPI
      RADIAN = PI/ONE80                                                 BLKDTA
      THIRD  = ONE/THREE                                                BLKDTA
      EPS(1) = ONE/TEN                                                  BLKDTA
      DO  10  I=2,24                                                    BLKDTA
   10 EPS(I) = EPS(I-1)/TEN                                             BLKDTA
      RETURN                                                            BLKDTA
      END                                                               BLKDTA
```

```
      SUBROUTINE CFACTT(A,B,D)                                          CFACTT
C                                                    REV 03    05/31/73CFACTT
C        GIVEN 3X3 MATRIX A                                             CFACTT
C        COMPUTE B TRANSPOSE OF COFACTORS (SIGNED MINORS)              CFACTT
C        AND D THE VALUE OF THE DETERMINANT OF A.                       CFACTT
C        INVERSE OF A IS B(J,K)/D                                       CFACTT
C                                                                       CFACTT
      IMPLICIT REAL*8 (A-H,O-Z)                                         CFACTT
      DIMENSION A(3,3),B(3,3)                                           CFACTT
      M = 4                                                             CFACTT
      L = 2                                                             CFACTT
      N = 3                                                             CFACTT
      D = 0.0                                                           CFACTT
      DO 20 J=1,3                                                       CFACTT
      B(J,J) = A(L,L)*A(N,N)-A(L,N)*A(N,L)                              CFACTT
      IF (J.EQ.3) GO TO 20                                              CFACTT
      L = N                                                             CFACTT
      N = J                                                             CFACTT
      KK = J+1                                                          CFACTT
      DO 15 K=KK,3                                                      CFACTT
      M = M-1                                                           CFACTT
      B(K,J) = A(K,M)*A(M,J)-A(K,J)*A(M,M)                              CFACTT
   15 B(J,K) = A(J,M)*A(M,K)-A(J,K)*A(M,M)                              CFACTT
   20 D = D+A(1,J)*B(J,1)                                               CFACTT
      RETURN                                                            CFACTT
      END                                                               CFACTT
```

```
      SUBROUTINE CHAIN(ISKIP)                                           JDRIFT
C                                                   REV IV    07/24/86SLIP
C     COMPUTES THE LINEAR POSITION AND VELOCITY IN INERTIAL REFERENCE   CHAIN
C     OF BODY SEGMENTS FROM THOSE OF THE REFERENCE SEGMENTS             CHAIN
C     (I.E., SEGMENT NO. 1 AND EACH SEGMENT J FOR WHICH JNT(J)=0).      CHAIN
C                                                                       CHAIN
      IMPLICIT REAL*8(A-H,O-Z)                                          CHAIN
      COMMON/CONTRL/ TIME,NSEG,NJNT,NPL,NBLT,NBAG,NVEH,NGRND,           CHAIN
     *               NS,NQ,NSD,NFLX,NHRNSS,NWINDF,NJNTF,NPRT(36),NPG    PAGE
      COMMON/SGMNTS/ D(3,3,30),WMEG(3,30),WMEGD(3,30),U1(3,30),U2(3,30),CHAIN
     *               SEGLP(3,30),SEGLV(3,30),SEGLA(3,30),NSYM(30)       CHAIN
      COMMON/DESCRP/ PHI(3,30),W(30),RW(30),SR(4,60),HA(3,60),HB(3,60), SLIP
     *               RPHI(3,30),HT(3,3,60),SPRING(5,90),VISC(7,90),     CHAIN
     *               JNT(30),IPIN(30),ISING(30),IGLOB(30),JOINTF(30)    CHAIN
      COMMON/CEULER/ IEULER(30),HIR(3,3,90),ANG(3,30),ANGD(3,30),       SLIP
     *               FE(3,30),TQE(3,30),CONST(5,30)                     SLIP
      COMMON/TEMPVS/ T1(3),T2(3),T3(3),T4(3),T5(3),T6(3),T7(3)          SLIP
      DATA IFIRST/1/                                                    SLIP
      CALL ELTIME (1,11)                                                ATBIII
      IF (NJNT.EQ.0) GO TO 71                                           ATBIII
      IF(ISKIP.NE.0) CALL DRIFT                                         JDRIFT
      DO 70 J=1,NJNT                                                    ATBIII
      K = IABS(JNT(J))                                                  ATBIII
      IF (K.EQ.0) GO TO 70                                              ATBIII
      IF (ISING(J+1).LT.0) GO TO 70                                     ATBIII
C                                                                       ATBIII
C     COMPUTE SEGMENT POSITIONS BY                                      ATBIII
C      P(J+1) = P(K) + D(K)'*R(K,J) - D(J+1)'*R(J+1,J)                  ATBIII
C                                                                       ATBIII
C     COMPUTE SEGMENT VELOCITIES BY                                     ATBIII
C      V(J+1) = V(K) + D(K)'*W(K) X R(K,J) - D(J+1)'*W(J+1) X R(J+1,J)  ATBIII
C                                                                       ATBIII
      CALL CROSS (WMEG(1,K),SR(1,2*J-1),T1)                             JDRIFT
      CALL DOT31 (D(1,1,K),T1,T3)                                       ATBIII
      CALL CROSS (WMEG(1,J+1),SR(1,2*J),T2)                             ATBIII
      CALL DOT31 (D(1,1,J+1),T2,T4)                                     ATBIII
      CALL DOT31 (D(1,1,K),SR(1,2*J-1),T1)                              ATBIII
      CALL DOT31 (D(1,1,J+1),SR(1,2*J),T2)                              ATBIII
      IF (IABS(IPIN(J)).LT.5) GO TO 50                                  SLIP
      IF (IEULER(J).EQ.-1)GO TO 50                                      SLIP
      IF (IFIRST.EQ.1) GO TO 50                                         SLIP
      DO 40 I = 1,3                                                     SLIP
      T5(I) = SEGLP(I,J+1) + T2(I) - SEGLP(I,K) - T1(I)                 SLIP
   40 T6(I) = SEGLV(I,J+1) + T4(I) - SEGLV(I,K) - T3(I)                 SLIP
      CALL DOT31 (D(1,1,K),HT(1,3,2*J-1),T7)                            SLIP
      SR(4,2*J-1) = T5(1)*T7(1) + T5(2)*T7(2) + T5(3)*T7(3)             SLIP
      SR(4,2*J  ) = T6(1)*T7(1) + T6(2)*T7(2) + T6(3)*T7(3)             SLIP
      CALL CROSS (WMEG(1,K),HT(1,3,2*J-1),T5)                           SLIP
      CALL DOT31 (D(1,1,K),T5,T6)                                       SLIP
      DO 45 I =1 ,3                                                     SLIP
```

```
      T1(I) = T1(I) + SR(4,2*J-1)*T7(I)                               SLIP
   45 T3(I) = T3(I) + SR(4,2*J  )*T7(I) + SR(4,2*J-1)*T6(I)           SLIP
   50 DO 60 I=1,3                                                     SLIP
      SEGLP(I,J+1) = SEGLP(I,K) + T1(I) - T2(I)                       ATBIII
   60 SEGLV(I,J+1) = SEGLV(I,K) + T3(I) - T4(I)                       ATBIII
   70 CONTINUE                                                        CHAIN
      IFIRST = 0                                                      SLIP
C                                                                     CHAIN
C     OPTIONAL OUTPUT                                                 CHAIN
C                                                                     CHAIN
   71 IF  (NPRT(20).NE.0)  WRITE(6,90)  TIME                          CHAIN
     *                          ,((SEGLP(I,J),I=1,3),J=1,NSEG)        CHAIN
     *                          ,((SEGLV(I,J),I=1,3),J=1,NSEG)        CHAIN
   90 FORMAT('0 LINEAR POSITIONS AND VELOCITIES OF BODY SEGMENTS FROM CHCHAIN
     *AIN FOR TIME =',F12.6/(9F13.5))                                 CHAIN
      CALL ELTIME(2,11)                                               CHAIN
      RETURN                                                          CHAIN
      END                                                             CHAIN
```

```
      SUBROUTINE CINPUT                                                 CINPUT
C                                                 REV III.2 08/08/84REVIII
C     INPUT CARDS E.1 - E.4 FOR THE FORCE-DEFLECTION, INERTIAL SPIKE,   CINPUT
C     R FACTOR, G FACTOR AND FRICTION COEFFICIENT FUNCTION DEFINITIONS  CINPUT
C                                                                       CINPUT
      IMPLICIT REAL*8(A-H,O-Z)                                          CINPUT
      COMMON/CONTRL/ TIME,NSEG,NJNT,NPL,NBLT,NBAG,NVEH,NGRND,           PAGE
     *                NS,NQ,NSD,NFLX,NHRNSS,NWINDF,NJNTF,NPRT(36),NPG   PAGE
      COMMON/TABLES/MXNTI,MXNTB,MXTB1,MXTB2,NTI(50),NTAB(1250),TAB(4500)DIMENB
      COMMON/TEMPVS/JTITLE(5,51),NF(5),NT(3),KTITLE(31)                 CINPUT
      REAL JTITLE,KTITLE                                                CINPUT
C                                                                       CINPUT
      IS = 0                                                            CINPUT
      DO 10 I = 1,50                                                    CINPUT
   10 NTI(I) = 0                                                        CINPUT
      J1 = 1                                                            CINPUT
C                                                                       CINPUT
C     INPUT CARD E.1 - FUNCTION NO. AND TITLE, IF NO. > 50 SKIP OUT.    CINPUT
C                                                                       CINPUT
   11 READ(5,12) I,(KTITLE(J  ),J = 1,5)                                CINPUT
   12 FORMAT (I4,4X,5A4)                                                CINPUT
      IF (I.GT.50) GO TO 30                                             CINPUT
      DO 13 J = 1,5                                                     CINPUT
   13 JTITLE(J,I) = KTITLE(J)                                           CINPUT
C                                                                       CINPUT
C     HAS FUNCTION NO. BEEN ALREADY USED?                               CINPUT
C                                                                       CINPUT
      IF  (NTI(I).NE.0)  WRITE(6,14) I                                  CINPUT
   14 FORMAT('0 FUNCTION NO.',I4,' HAS ALREADY BEEN INPUTTED AND WILL BECINPUT
     *REPLACED BY NEXT FUNCTION')                                       CINPUT
      NTI(I) = J1                                                       CINPUT
      J2 = J1+4                                                         CINPUT
C                                                                       CINPUT
C     INPUT CARD E.2                                                    CINPUT
C                                                                       CINPUT
      READ(5,15) (TAB(J),J = J1,J2)                                     CINPUT
   15 FORMAT (6F12.0)                                                   CINPUT
      IS = 1-IS                                                         CINPUT
      IF (IS.EQ.0) WRITE(6,16)                                          CINPUT
      IF (IS.EQ.0) GOTO 40                                              PAGE
      WRITE(6,41) NPG                                                   PAGE
   41 FORMAT('1',122X,'PAGE',I5)                                        PAGE
      NPG=NPG+1                                                         PAGE
   16 FORMAT(/////)                                                     CINPUT
   40 WRITE(6,17) I,(JTITLE(J,I),J=1,5),I,NTI(I),(TAB(J),J=J1,J2)       PAGE
   17 FORMAT(' FUNCTION NO.',I4,4X,5A4,20X,'NTI(',I2,') =',I5,45X,      PAGE
     * 'CARDS E'//10X,'D0',13X,'D1',13X,'D2',13X,'D3',13X,'D4'/5F15.4//)CINPUT
      D0 = TAB(J1)                                                      CINPUT
      D1 = TAB(J1+1)                                                    CINPUT
      D2 = TAB(J1+2)                                                    CINPUT
```

```
      J1 = J2+1                                                         CINPUT
      IF (D1)  22,18,20                                                 CINPUT
C                                                                       CINPUT
C     FUNCTION IS CONSTANT  D2 FOR ALL D.                               CINPUT
C                                                                       CINPUT
   18 WRITE(6,19)  D2                                                   CINPUT
   19 FORMAT(7X,'FUNCTION IS CONSTANT',F12.6)                           CINPUT
      GO TO 11                                                          CINPUT
C                                                                       CINPUT
C       5TH ORDER POLYNOMIAL ...  1ST FUNCTION                          CINPUT
C       INPUT CARD E.3                                                  CINPUT
C                                                                       CINPUT
   20 J2 = J1+5                                                         CINPUT
      READ(5,15)(TAB(J),J = J1,J2)                                      CINPUT
      WRITE(6,21)  (TAB(J),J = J1,J2)                                   CINPUT
   21 FORMAT(7X,'FIRST PART OF FUNCTION - 5TH DEGREE POLYNOMIAL'//      CINPUT
     *         8X,'A0',13X,'A1',13X,'A2',13X,'A3',13X,'A4',13X,'A5',13X/CINPUT
     *         6F15.6//)                                                CINPUT
      J1 = J2+1                                                         CINPUT
      GO TO 25                                                          CINPUT
C                                                                       CINPUT
C         TABLE LOAD ...  1ST FUNCTION                                  CINPUT
C         INPUT CARDS E.4.A-E.4.N                                       CINPUT
C                                                                       CINPUT
   22 READ(5,23)  NPI                                                   CINPUT
   23 FORMAT (12I6)                                                     CINPUT
      TAB(J1) = NPI                                                     CINPUT
      J1 = J1+1                                                         CINPUT
      J2 = J1+2*NPI-1                                                   CINPUT
      READ(5,15)(TAB(J),J = J1,J2)                                      CINPUT
      WRITE(6,24)  NPI, (TAB(J)    ,J = J1, J2)                         CINPUT
   24 FORMAT(7X,'FIRST PART OF FUNCTION - ',I4,' TABULAR POINTS'//      CINPUT
     *         8X,'D',16X,'F(D)'  /(F15.6,F15.4))                       CINPUT
      J1 = J2+1                                                         CINPUT
C                                                                       CINPUT
C        CHECK FOR SECOND FUNCTION                                      CINPUT
C                                                                       CINPUT
   25 IF(D2)  28,11,26                                                  CINPUT
C                                                                       CINPUT
C       SECOND FUNCTION ...  5TH ORDER POLYNOMIAL                       CINPUT
C       INPUT CARD E.3                                                  CINPUT
C                                                                       CINPUT
   26 J2 = J1+5                                                         CINPUT
      READ(5,15)(TAB(J),J = J1,J2)                                      CINPUT
      WRITE(6,27)  (TAB(J),J = J1,J2)                                   CINPUT
   27 FORMAT(7X,'SECOND PART OF FUNCTION - 5TH DEGREE POLYNOMIAL'//     CINPUT
     *         8X,'B0',13X,'B1',13X,'B2',13X,'B3',13X,'B4',13X,'B5',13X/CINPUT
     *         6F15.6//)                                                CINPUT
      J1 = J2+1                                                         CINPUT
      GO TO 11                                                          CINPUT
```

```
C                                                                     CINPUT
C       SECOND FUNCTION ...  TABLE LOAD                               CINPUT
C       INPUT CARDS E.4.A-E.4.N                                       CINPUT
C                                                                     CINPUT
   28 READ(5,23)  NPI                                                 CINPUT
      TAB(J1) = NPI                                                   CINPUT
      J1 = J1+1                                                       CINPUT
      J2 = J1+2*NPI-1                                                 CINPUT
      READ(5,15)(TAB(J),J = J1,J2)                                    CINPUT
      WRITE(6,29) NPI, (TAB(J), J = J1,J2)                            CINPUT
   29 FORMAT(7X,'SECOND PART OF FUNCTION - ',I4,' TABULAR POINTS'//   CINPUT
     *         8X,'D',16X,'F(D)'  /(F15.6,F15.4))                     CINPUT
      J1 = J2+1                                                       CINPUT
      GO TO 11                                                        CINPUT
   30 MXTB1 = J1-1                                                    CINPUT
      CALL KINPUT                                                     CINPUT
      CALL FINPUT                                                     CINPUT
      CALL HINPUT                                                     CINPUT
      RETURN                                                          CINPUT
      END                                                             CINPUT
```

```
      SUBROUTINE CMPUTE (K,M,FT)                                        CMPUTE
C                                                   REV III.2 08/08/84REVIII
      IMPLICIT REAL*8 (A-H,O-Z)                                         CMPUTE
      COMMON/CONTRL/ TIME,NSEG,NJNT,NPL,NBLT,NBAG,NVEH,NGRND,           CMPUTE
     *               NS,NQ,NSD,NFLX,NHRNSS,NWINDF,NJNTF,NPRT(36),NPG    PAGE
      COMMON/CDINT/ UU(4),GH(3,4),                                      CMPUTE
     *              E(3,240), F(5,240),GG(5,240),Y(5,240),U(5,240),     CMPUTE
     *              H,HPRINT,HS,TPRINT,TSTART,ICNT,IDBL,IFLAG           CMPUTE
      COMMON/COMAIN/ VAR(240),DER(240),DT,HO,HMAX,HMIN,RSTIME,          CMPUTE
     *               ISTEP,NSTEPS,NDINT,NEQ,IRSIN,IRSOUT                CMPUTE
      TIME = TSTART + FT                                                CMPUTE
      CALL DZP (NEQ,VAR,GG,E,FT,M)                                      CMPUTE
      IF  (NPRT(26).EQ.2)  CALL OUTPUT(0)                               CMPUTE
      CALL PDAUX (VAR,DER,NEQ,K)                                        CMPUTE
      IF  (NPRT(26).EQ.2)  CALL OUTPUT(1)                               CMPUTE
      RETURN                                                            CMPUTE
      END                                                               CMPUTE
```

```
      SUBROUTINE CONTCT                                                 CONTCT
C                                                   REV III.2 08/08/84REVIII
C     CONTROLS THE CALLING OF SUBROUTINES REQUIRED TO COMPUTE THOSE     CONTCT
C     EXTERNAL FORCES AND TORQUES ACTING ON THE BODY SEGMENTS.          CONTCT
C                                                                       CONTCT
      IMPLICIT REAL*8 (A-H,O-Z)                                         CONTCT
      COMMON/CONTRL/ TIME,NSEG,NJNT,NPL,NBLT,NBAG,NVEH,NGRND,           CONTCT
     *               NS,NQ,NSD,NFLX,NHRNSS,NWINDF,NJNTF,NPRT(36),NPG    PAGE
      COMMON/JBARTZ/ MNPL(   30),MNBLT(   8),MNSEG(   30),MNBAG(    6), CONTCT
     *               MPL(3,5,30),MBLT(3,5,8),MSEG(3,5,30),MBAG(3,10,6), CONTCT
     *               NTPL( 5,30),NTBLT( 5,8),NTSEG( 5,30)               CONTCT
      COMMON/FORCES/PSF(7,70),BSF(4,20),SSF(10,40),BAGSF(3,20),         NCFORC
     *               PRJNT(7,30),NPANEL(5),NPSF,NBSF,NSSF,NBGSF         CONTCT
      COMMON/TABLES/MXNTI,MXNTB,MXTB1,MXTB2,NTI(50),NTAB(1250),TAB(4500)DIMENB
      COMMON/HRNESS/ BAR(15,100),BB(100),BBDOT(100),PLOSS(2,100),       CONTCT
     *               XLONG(20),HTIME(2),IBAR(5,100),NL(2,100),          CONTCT
     *               NPTSPB(20),NPTPLY(20),NTHRNS(20),NBLTPH(5)         CONTCT
      COMMON/WINDFR/ WTIME(30),QFU(3,5),QFV(3,5),WF(3,30),IWIND(30),    WINDOP
     *               MWSEG(7,30),NFVSEG(6),NFVNT(5),MOWSEG(30,30)       WINDOP
      DATA MAXPSF/70/,MAXBSF/20/,MAXSSF/46/                             NCFORC
C                                                                       CHGIII
C     MAXSSF SHOULD BE 40 BUT IT IS ALLOWED TO OVERFLOW INTO BAGSF      NCFORC
C                                                                       CHGIII
      CALL ELTIME(1,12)                                                 CONTCT
      NPSF = 0                                                          CONTCT
      NBSF = 0                                                          CONTCT
      NSSF = 0                                                          CONTCT
      IF (NPL.LE.0) GO TO 21                                            CONTCT
C                                                                       CONTCT
C     CALL PLELP ROUTINE FOR EACH ALLOWED PLANE-SEGMENT CONTACT.        CONTCT
C                                                                       CONTCT
      DO 20 J=1,NPL                                                     CONTCT
      IF(MNPL(J).EQ.0) GO TO 20                                         CONTCT
      KPL = MNPL(J)                                                     CONTCT
      DO 19 I=1,KPL                                                     CONTCT
      NPSF = NPSF+1                                                     CONTCT
      IF(NPSF.GT.MAXPSF) STOP 57                                        CHGIII
      M1 = MPL(1,I,J)                                                   CONTCT
      M2 = MPL(2,I,J)                                                   CONTCT
      M3 = MPL(3,I,J)                                                   CONTCT
      NT = NTPL(I,J)                                                    CONTCT
      JT = NTAB(NT)                                                     CONTCT
      TAB(JT) = 0.0                                                     CONTCT
   19 CALL PLELP(M2,M3,M1,J,NT)                                         CONTCT
   20 CONTINUE                                                          CONTCT
   21 IF(NBLT.LE.0) GO TO 41                                            CONTCT
C                                                                       CONTCT
C     CALL BELTRT ROUTINE FOR EACH ALLOWED BELT-SEGMENT CONTACT.        CONTCT
C                                                                       CONTCT
      DO 30 J=1,NBLT                                                    CONTCT
```

```
      IF(MNBLT(J).EQ.0) GO TO 30                                        CONTCT
      KBLT = MNBLT(J)                                                   CONTCT
      DO 29 I=1,KBLT                                                    CONTCT
      NBSF = NBSF+1                                                     CONTCT
      IF(NBSF.GT.MAXBSF) STOP 58                                        CHGIII
      M1 = MBLT(1,I,J)                                                  CONTCT
      M2 = MBLT(2,I,J)                                                  CONTCT
      M3 = MBLT(3,I,J)                                                  CONTCT
      NT = NTBLT(I,J)                                                   CONTCT
      JT = NTAB(NT)                                                     CONTCT
      TAB(JT) = 0.0                                                     CONTCT
      NF = NTAB(NT+5)                                                   CONTCT
      IF (NF.NE.0) JT = NTAB(NT+6)                                      CONTCT
      IF (NF.NE.0) TAB(JT) = 0.0                                        CONTCT
   29 CALL BELTRT(M2,M3,M1,J,NT)                                        CONTCT
   30 CONTINUE                                                          CONTCT
C                                                                       CONTCT
C     CALL SEGSEG ROUTINE FOR EACH ALLOWED SEGMENT-SEGMENT CONTACT.     CONTCT
C                                                                       CONTCT
   41 DO 50 J=1,NSEG                                                    CONTCT
      IF(MNSEG(J).EQ.0) GO TO 50                                        CONTCT
      KSEG = MNSEG(J)                                                   CONTCT
      DO 49 I=1,KSEG                                                    CONTCT
      NSSF = NSSF+1                                                     CONTCT
      IF(NSSF.GT.MAXSSF) STOP 59                                        CHGIII
      M1 = MSEG(1,I,J)                                                  CONTCT
      M2 = MSEG(2,I,J)                                                  CONTCT
      M3 = MSEG(3,I,J)                                                  CONTCT
      NT = NTSEG(I,J)                                                   CONTCT
      JT = NTAB(NT)                                                     CONTCT
      TAB(JT) = 0.0                                                     CONTCT
   49 CALL SEGSEG(J,M1,M2,M3,NT)                                        CONTCT
   50 CONTINUE                                                          CONTCT
C                                                                       CONTCT
C     CALL AIRBAG ROUTINE FOR ALLOWED BAG-SEGMENT CONTACTS, IF ANY.     CONTCT
C                                                                       CONTCT
      IF (NBAG.NE.0) CALL AIRBAG                                        CONTCT
C                                                                       CONTCT
C     CALL WINDY ROUTINE FOR WIND FORCES ON EACH SEGMENT.               CONTCT
C                                                                       CONTCT
      DO 60 J=1,NSEG                                                    CONTCT
      IF (MWSEG(1,J).EQ.0) GO TO 60                                     CONTCT
      M=MWSEG(1,J)                                                      WINDOP
      M1 = MWSEG(2,J)                                                   CONTCT
      M2 = MWSEG(3,J)                                                   CONTCT
      M3 = MWSEG(4,J)                                                   CONTCT
      NT = MWSEG(5,J)                                                   CONTCT
      CALL WINDY (M,M1,M2,M3,NT)                                        WINDOP
   60 CONTINUE                                                          CONTCT
C                                                                       CONTCT
```

```
C      CALL WINDY FOR FORCE FUNCE FUNCTION CALCULATIONS.                  CONTCT
C                                                                         CONTCT
       NFORCE = NFVSEG(6)                                                 CONTCT
       IF  (NFORCE.GT.0)  CALL WINDY (0,M1,M2,M3,NT)                      WINDOP
C                                                                         CONTCT
C      CALL HBELT ROUTINE FOR EACH HARNESS-BELT SYSTEM.                   CONTCT
C                                                                         CONTCT
       IF  (NHRNSS.LE.0)  GO TO 80                                        CONTCT
       J1 = 1                                                             CONTCT
       KNLO = 0                                                           CONTCT
       DO  70  I=1,NHRNSS                                                 CONTCT
       IF  (NBLTPH(I).LE.0)  GO TO 70                                     CONTCT
       J2 = J1 + NBLTPH(I) - 1                                            CONTCT
       CALL HBELT (J1,J2,KNLO,0)                                          CONTCT
       J1 = J2+1                                                          CONTCT
    70 CONTINUE                                                           CONTCT
C                                                                         CONTCT
C      CALL SPDAMP FOR SPRING DAMPER FORCES, IF ANY                       CONTCT
C                                                                         CONTCT
    80 IF  (NSD.NE.0)  CALL SPDAMP                                        CONTCT
       CALL ELTIME (2,12)                                                 CONTCT
       RETURN                                                             CONTCT
       END                                                                CONTCT
```

```
      SUBROUTINE CROSS(A,B,C)                                           CROSS
C                                                   REV 03    05/31/73CROSS
C     COMPUTES VECTOR CROSS PRODUCT  C = A X B.                         CROSS
C                                                                       CROSS
C       ARGUMENTS                                                       CROSS
C         A,B,C: VECTORS OF LENGTH 3 WHERE C=AXB.                       CROSS
C                                                                       CROSS
      IMPLICIT REAL*8 (A-H,O-Z)                                         CROSS
      DIMENSION A(3),B(3),C(3)                                          CROSS
       C(1) = A(2)*B(3) - A(3)*B(2)                                     CROSS
       C(2) = A(3)*B(1) - A(1)*B(3)                                     CROSS
       C(3) = A(1)*B(2) - A(2)*B(1)                                     CROSS
      RETURN                                                            CROSS
      END                                                               CROSS
```

```
      SUBROUTINE DAUX(I1)                                               DAUX
C                                                 REV IV    07/24/86SLIP
C     COMPUTES DERIVATIVES FOR INTEGRATOR ROUTINE BY                    DAUX
C      (1) SET UP INITIAL VALUES FOR ARRAY OF SYSTEM EQUATIONS.         DAUX
C      (2) MODIFY ARRAYS BY CONSTRAINTS.                                DAUX
C      (3) SOLVE SYSTEM OF EQUATION FOR F,TQ,QQ AND V4.                 DAUX
C      (4) EVALUATE DERIVATIVES SEGLA AND WMEGD.                        DAUX
C                                                                       DAUX
      IMPLICIT REAL*8(A-H,O-Z)                                          DAUX
      COMMON/CONTRL/ TIME,NSEG,NJNT,NPL,NBLT,NBAG,NVEH,NGRND,           DAUX
     *               NS,NQ,NSD,NFLX,NHRNSS,NWINDF,NJNTF,NPRT(36),NPG    PAGE
      COMMON/SGMNTS/ D(3,3,30),WMEG(3,30),WMEGD(3,30),U1(3,30),U2(3,30),DAUX
     *               SEGLP(3,30),SEGLV(3,30),SEGLA(3,30),NSYM(30)       DAUX
      COMMON/DESCRP/ PHI(3,30),W(30),RW(30),SR(4,60),HA(3,60),HB(3,60), SLIP
     *               RPHI(3,30),HT(3,3,60),SPRING(5,90),VISC(7,90),     DAUX
     *               JNT(30),IPIN(30),ISING(30),IGLOB(30),JOINTF(30)    DAUX
      COMMON/CMATRX/ V1(3,30),V2(3,30),V3(3,12),B12(3,3,60),A22(3,3,60),DAUX
     *               F(3,30),TQ(3,30),WJ(30),A11(3,3,30)                SLIP
      COMMON/CSTRNT/ A13(3,3,24),A23(3,3,24),B31(3,3,24),B32(3,3,24),   DAUX
     *               HHT(3,3,12),RK1(3,12),RK2(3,12),QQ(3,12),TQQ(3,12),DAUX
     *               RQQ(3,12),HQQ(3,12),SQQ(12),CFQQ(12),              DAUX
     *               KQ1(12),KQ2(12),KQTYPE(12)                         DAUX
      COMMON/FLXBLE/ HF(4,12,8),B42(3,3,24),V4(3,8),NFLEX(3,8)          DAUX
      COMMON/CNSNTS/ PI,RADIAN,G,THIRD,EPS(24),                         DAUX
     *               UNITL,UNITM,UNITT,GRAVTY(3),TWOPI                  TWOPI
      COMMON/RSAVE/  XSG(3,20,3),DPMI(3,3,30),LPMI(30),                 ATBIII
     *               NSG(9),MSG(20,9),MCG,MCGIN(24,5),KREF(20,9)        TTHKREF
C                                                                       DAUX
C     NOTE: DAUX SHARES /TEMPVS/ WITH DAUX11,12,22,31,32 &33.           DAUX
C                                                                       DAUX
      LOGICAL*1 FREE                                                    SLIP
      COMMON/TEMPVS/ C(3,3,600),RHS(3,54),IJK(54,54),IJ,NQ2S,           SLIP
     *               IDUM(458),FREE(30)                                 SLIP
      DIMENSION T1(3),T2(3),T3(3)                                       TGMOD2
      CALL ELTIME(1,9)                                                  DAUX
C                                                                       DAUX
C     IF I1#0, U1 AND U2 HAVE BEEN SET UP BY CALLING ROUTINE.           DAUX
C                                                                       DAUX
      IF (I1.NE.0) GO TO  8                                             DAUX
C                                                                       DAUX
C     SET UP INITIAL VALUES OF A & B ARRAYS AND U & V VECTORS.          DAUX
C     MODIFY U1 & U2 ARRAYS BY CONTACT AND JOINT FORCES.                DAUX
C                                                                       DAUX
      CALL CHAIN(NPRT(36))                                              JDRIFT
      CALL SETUP1                                                       DAUX
      CALL VEHPOS                                                       DAUX
      CALL CONTCT                                                       DAUX
      CALL VISPR(0,0)                                                   DAUX
      CALL EJOINT(0,0)                                                  DAUX
      CALL SETUP2                                                       DAUX
```

```
      IF (NFLX.GT.0) CALL FLXSEG                                       DAUX
C                                                                      DAUX
C     MODIFY U1,U2 AND ADD G TO U1.                                    DAUX
C                                                                      DAUX
      DO  5  J=1,NGRND                                                 DAUX
      IF  (ISING(J))  1,3,5                                            DAUX
    1 DO  2  I=1,3                                                     DAUX
      U1(I,J) = SEGLA(I,J)                                             DAUX
    2 U2(I,J) = WMEGD(I,J)                                             DAUX
      GO TO 5                                                          DAUX
    3 DO  4  I=1,3                                                     DAUX
      U1(I,J) = U1(I,J)*RW(J) + GRAVTY(I)                              DAUX
    4 U2(I,J) = U2(I,J)*RPHI(I,J)                                      DAUX
    5 CONTINUE                                                         DAUX
C                                                                      DAUX
C     SET UP BODY SEGMENT SYMMETRY                                     DAUX
C         NSYM(J) = 0    3D MOTION                                     DAUX
C         NSYM(J) = J    CENTRAL SEGMENT 2D MOTION, NO LATERAL MOTION  DAUX
C         NSYM(J) = K    SEGMENT J SYMMETRIC TO SEGMENT K, ALL MOTION  DAUX
C                        IN THE X-Z PLANE, NO LATERAL MOTION           DAUX
C         NSYM(J) = -K   SEGMENT J MIRROR SYMMETRIC TO SEGMENT K, EQUAL DAUX
C                        BUT OPPOSITE LATERAL MOTION PERMITTED         DAUX
C                                                                      DAUX
      DO  20  J=1,NGRND                                                DAUX
      IF (NSYM(J).EQ.0) GO TO 20                                       DAUX
      K = IABS(NSYM(J))                                                DAUX
      DO 205 L=1,3                                                     TGMOD2
      T1(L) = U2(L,J)                                                  TGMOD2
      T2(L) = U2(L,K)                                                  TGMOD2
      T3(L) = U2(L,J)                                                  TGMOD2
  205 CONTINUE                                                         TGMOD2
      IF(LPMI(J).EQ.0.AND.LPMI(K).EQ.0) GO TO 201                      TGMOD2
      IF(LPMI(J).NE.0.AND.LPMI(K).EQ.0) GO TO 202                      TGMOD2
      IF(LPMI(J).EQ.0.AND.LPMI(K).NE.0) GO TO 203                      TGMOD2
      CALL DOT31(DPMI(1,1,J),U2(1,J),T1)                               TGMOD2
      CALL DOT31(DPMI(1,1,K),U2(1,K),T2)                               TGMOD2
      GO TO 201                                                        TGMOD2
  202 CALL DOT31(DPMI(1,1,J),U2(1,J),T1)                               TGMOD2
      GO TO 201                                                        TGMOD2
  203 CALL DOT31(DPMI(1,1,K),U2(1,K),T2)                               TGMOD2
  201 CONTINUE                                                         TGMOD2
      IF (NSYM(J).EQ.J) GO TO 19                                       DAUX
      IF (K.LT.J) GO TO 16                                             DAUX
      U1(1,J) = 0.5*(U1(1,J) + U1(1,K))                                DAUX
      U1(3,J) = 0.5*(U1(3,J) + U1(3,K))                                DAUX
      T3(2) = 0.5*(T1(2) + T2(2))                                      TGMOD2
      GO TO 17                                                         DAUX
   16 U1(1,J) = U1(1,K)                                                DAUX
      U1(3,J) = U1(3,K)                                                DAUX
      T3(2) = T2(2)                                                    DAUX
```

```
   17 IF (NSYM(J).GT.0) GO TO 19                                        DAUX
      IF (K.LT.J) GO TO 18                                              DAUX
      U1(2,J) = 0.5*(U1(2,J) - U1(2,K))                                 DAUX
      T3(1) = 0.5*(T1(1) - T2(1))                                       TGMOD2
      T3(3) = 0.5*(T1(3) - T2(3))                                       TGMOD2
      GO TO 206                                                         DAUX
   18 U1(2,J) = -U1(2,K)                                                DAUX
      T3(1) = -T2(1)                                                    TGMOD2
      T3(3) = -T2(3)                                                    TGMOD2
      GO TO 206                                                         DAUX
   19 U1(2,J) = 0.0                                                     DAUX
      T3(1) = 0.0                                                       TGMOD2
      T3(3) = 0.0                                                       TGMOD2
  206 IF(LPMI(J).EQ.0) GO TO 207                                        TGMOD2
      CALL MAT31(DPMI(1,1,J),T3,U2(1,J))                                TGMOD2
      GO TO 20                                                          TGMOD2
  207 U2(1,J) = T3(1)                                                   TGMOD2
      U2(2,J) = T3(2)                                                   TGMOD2
      U2(3,J) = T3(3)                                                   TGMOD2
   20 CONTINUE                                                          TGMOD2
C                                                                       DAUX
C     INITIALIZE IJK ARRAY AND IJ COUNTER TO ZERO.                      DAUX
C                                                                       DAUX
    8 NQ2S = 2*NS + NFLX + NQ                                           DAUX
      NJ2  = NQ2S + 2*NJNT                                              DAUX
      IF (NJ2.GT.54) WRITE (6,11) NS,NFLX,NQ,NJNT,NJ2                   DAUX
   11 FORMAT('0NS=',I6,',NFLX=',I6,',NQ=',I6,',NJNT=',I6,' AND NJ2=',I6/AFREVS
     *' THE VALUE OF NJ2 EXCEEDS THE ARRAY SIZES FOR RHS AND IJK IN SUBRDAUX
     *OUTINE DAUX. PROGRAM TERMINATED.')                                DAUX
      IF  (NJ2.GT.54)  STOP 34                                          DAUX
      MJ2 = NJ2                                                         DAUX
      DO 10 I=1,NJ2                                                     DAUX
      DO 10 J=1,NJ2                                                     DAUX
   10 IJK(I,J) = 0                                                      DAUX
      IJ = 0                                                            DAUX
C                                                                       DAUX
C     ELMINATE SEGLA AND WMEGD FROM SYSTEM OF EQUATIONS.                DAUX
C                                                                       DAUX
      IF (NS.GT.0)   CALL DAUX55                                        DAUX
      IF (NJNT.EQ.0) GO TO 12                                           DAUX
      IF (NFLX.GT.0) CALL DAUX44                                        DAUX
      CALL DAUX11                                                       DAUX
      CALL DAUX12                                                       DAUX
      CALL DAUX22                                                       DAUX
   12 IF (NQ.LE.0) GO TO 15                                             DAUX
      IF (NJNT.EQ.0) GO TO 13                                           DAUX
      CALL DAUX31                                                       DAUX
      CALL DAUX32                                                       DAUX
   13 CALL DAUX33                                                       DAUX
      DO 14 I=1,NQ                                                      DAUX
```

```
   14 IF (KQTYPE(I).GE.4) MJ2 = -NJ2                                DAUX
   15 IF (NPRT(8).EQ.0) GO TO 28                                    DAUX
   21 WRITE (6,22) NPG,(J,J=1,NJ2)                                  PAGE
      NPG=NPG+1                                                     PAGE
   22 FORMAT('1 DAUX PRINT OF IJK MATRIX',97X,'PAGE',I5//6X,40I3)   PAGE
      DO 23 I=1,NJ2                                                 DAUX
   23 WRITE (6,24) I,(IJK(I,J),J=1,NJ2)                             DAUX
   24 FORMAT(I3,3X,40I3)                                            DAUX
      WRITE (6,29)                                                  DAUX
   29 FORMAT('0 DAUX PRINT OF RHS ARRAY'//)                         DAUX
      DO 30 K=1,NJ2                                                 DAUX
   30 WRITE (6,27) K,(RHS(I,K),I=1,3)                               DAUX
      WRITE (6,25) NPG                                              PAGE
      NPG=NPG+1                                                     PAGE
   25 FORMAT('1 DAUX PRINT OF C ARRAY ELEMENTS',91X,'PAGE',I5//)    PAGE
      DO 26 K=1,IJ                                                  DAUX
   26 WRITE (6,27) K,((C(I,J,K),J=1,3),I=1,3)                       DAUX
   27 FORMAT(I6,9G14.7)                                             DAUX
   28 IF (NPRT(8).EQ.-2) GO TO 31                                   DAUX
C                                                                   DAUX
C     SOLVE SYSTEM OF EQUATIONS FOR F,TQ,QQ & V4.                   DAUX
C                                                                   DAUX
      CALL FSMSOL (C,RHS,IJK,MJ2,IJ,54,600)                         CHGIII
      IF (NPRT(8).EQ. 2) NPRT(8) = -2                               DAUX
      IF (NPRT(8).EQ.-2) GO TO 21                                   DAUX
   31 IF (NPRT(8).EQ.-2) NPRT(8) = 0                                DAUX
      EPS12 = EPS(12)                                               JDRIFT
      IF (NJNT.EQ.0) GO TO 49                                       DAUX
      DO 51 I=1,NJNT                                                DAUX
      NJ = NQ2S + I                                                 DAUX
      NI = NJ+NJNT                                                  DAUX
      DO 51 K=1,3                                                   DAUX
      IF (DABS(RHS(K,NJ)).LT.EPS12) RHS(K,NJ) = 0.0                 DAUX
      IF (DABS(RHS(K,NI)).LT.EPS12) RHS(K,NI) = 0.0                 DAUX
      TQ(K,I) = TQ(K,I) - RHS(K,NI)                                 DAUX
   51 F(K,I) = RHS(K,NJ)                                            DAUX
   49 IF (NQ.EQ.0) GO TO 53                                         DAUX
      DO 52 I=1,NQ                                                  DAUX
      J = 2*NS + NFLX + I                                           DAUX
      DO 52 K=1,3                                                   DAUX
      IF (KQTYPE(I).LT.0)          RHS(K,J) = 0.0                   DAUX
      IF (DABS(RHS(K,J)).LT.EPS12) RHS(K,J) = 0.0                   DAUX
   52 QQ(K,I) = RHS(K,J)                                            DAUX
   53 IF (NFLX.EQ.0) GO TO 70                                       DAUX
      DO 54 I=1,NFLX                                                DAUX
      J = 2*NS + I                                                  DAUX
      DO 54 K=1,3                                                   DAUX
      IF (DABS(RHS(K,J)).LT.EPS12) RHS(K,J) = 0.0                   DAUX
   54 V4(K,I) = RHS(K,J)                                            DAUX
C                                                                   DAUX
```

```
C     BACKUP SOLUTION FOR SEGLA AND WMEGD.                                   DAUX
C                                                                            DAUX
   70 DO  71  J=1,NGRND                                                      DAUX
      DO  71  I=1,3                                                          DAUX
      SEGLA(I,J) = U1(I,J)                                                   DAUX
   71 WMEGD(I,J) = U2(I,J)                                                   DAUX
      IF (NS.EQ.0) GO TO 79                                                  DAUX
C                                                                            DAUX
C     SET UP SEGLA & WMEGD FOR SINGULAR SEGMENTS.                            DAUX
C                                                                            DAUX
      IS = 0                                                                 DAUX
      DO 78 J=1,NGRND                                                        DAUX
      IF (ISING(J).LE.0) GO TO 78                                            DAUX
      IS = IS+2                                                              DAUX
      DO 77 I=1,3                                                            DAUX
      IF (DABS(RHS(I,IS-1)).LT.EPS12) RHS(I,IS-1) = 0.0                      DAUX
      SEGLA(I,J) = SEGLA(I,J) + RHS(I,IS-1)                                  DAUX
      IF (DABS(RHS(I,IS  )).LT.EPS12) RHS(I,IS  ) = 0.0                      DAUX
   77 WMEGD(I,J) = WMEGD(I,J) + RHS(I,IS)                                    DAUX
   78 CONTINUE                                                               DAUX
   79 IF (NJNT.EQ.0) GO TO 80                                                DAUX
C                                                                            DAUX
C     ELIMINATE F                                                            DAUX
C                                                                            DAUX
      DO 75 M=1,NJNT                                                         DAUX
      N = IABS(JNT(M))                                                       DAUX
      IF (N.EQ.0) GO TO 73                                                   DAUX
      DO 72 I=1,3                                                            DAUX
      DO 72 J=1,3                                                            DAUX
      SEGLA(I,N  ) = SEGLA(I,N  ) - A11(I,J,M)*RW(N  )*F(J,M)                SLIP
      SEGLA(I,M+1) = SEGLA(I,M+1) + A11(I,J,M)*RW(M+1)*F(J,M)                SLIP
      WMEGD(I,N)   = WMEGD(I,N  ) - B12(J,I,2*M-1)*RPHI(I,N  )*F(J,M)        DAUX
   72 WMEGD(I,M+1) = WMEGD(I,M+1) - B12(J,I,2*M  )*RPHI(I,M+1)*F(J,M)        DAUX
C                                                                            DAUX
C     ELIMINATE TQ                                                           DAUX
C                                                                            DAUX
   73 IF (FREE(M)) GO TO 75                                                  SLIP
      L = NQ2S + NJNT + M                                                    DAUX
      DO 74 I=1,3                                                            DAUX
      DO 74 J=1,3                                                            DAUX
      WMEGD(I,N  ) = WMEGD(I,N  ) - A22(I,J,2*M-1)*RPHI(I,N  )*RHS(J,L)      DAUX
   74 WMEGD(I,M+1) = WMEGD(I,M+1) + A22(I,J,2*M  )*RPHI(I,M+1)*RHS(J,L)      DAUX
   75 CONTINUE                                                               DAUX
   80 IF (NQ.EQ.0) GO TO 83                                                  DAUX
C                                                                            DAUX
C     ELIMINATE QQ                                                           DAUX
C                                                                            DAUX
      DO 82 K=1,NQ                                                           DAUX
      IF (KQTYPE(K).LT.0) GO TO 82                                           DAUX
      N = KQ1(K)                                                             DAUX
```

```
      M = KQ2(K)                                                        DAUX
      DO 81 I=1,3                                                       DAUX
      DO 81 J=1,3                                                       DAUX
      SEGLA(I,N) = SEGLA(I,N) - A13(I,J,2*K-1)*RW(N)    *QQ(J,K)        DAUX
      SEGLA(I,M) = SEGLA(I,M) - A13(I,J,2*K  )*RW(M)    *QQ(J,K)        DAUX
      WMEGD(I,N) = WMEGD(I,N) - A23(I,J,2*K-1)*RPHI(I,N)*QQ(J,K)        DAUX
   81 WMEGD(I,M) = WMEGD(I,M) - A23(I,J,2*K  )*RPHI(I,M)*QQ(J,K)        DAUX
   82 CONTINUE                                                          DAUX
   83 IF (NFLX.EQ.0) GO TO 90                                           DAUX
C                                                                       DAUX
C     ELIMINATE V4 (TORQUES FOR FLEXIBLE SEGMENTS)                      DAUX
C                                                                       DAUX
      DO 84 N=1,NFLX                                                    DAUX
      N1 = NFLEX(1,N)                                                   DAUX
      N2 = NFLEX(2,N)                                                   DAUX
      N3 = NFLEX(3,N)                                                   DAUX
      DO 84 I=1,3                                                       DAUX
      DO 84 J=1,3                                                       DAUX
      WMEGD(I,N1) = WMEGD(I,N1) - B42(J,I,3*N-2)*RPHI(I,N1)*V4(J,N)     DAUX
      WMEGD(I,N2) = WMEGD(I,N2) - B42(J,I,3*N-1)*RPHI(I,N2)*V4(J,N)     DAUX
   84 WMEGD(I,N3) = WMEGD(I,N3) - B42(J,I,3*N  )*RPHI(I,N3)*V4(J,N)     DAUX
   90 DO 91 J=1,NGRND                                                   DAUX
      DO 91 I=1,3                                                       DAUX
      IF (DABS(WMEGD(I,J)).LE.EPS12) WMEGD(I,J) = 0.0                   DAUX
   91 IF (DABS(SEGLA(I,J)).LE.EPS12) SEGLA(I,J) = 0.0                   DAUX
C                                                                       DAUX
C     OPTIONAL OUTPUT OF FUNCTIONS AND DERIVATIVES.                     DAUX
C                                                                       DAUX
      IF (NPRT(9).NE.0) CALL PRINT(6H DAUX )                            DAUX
C                                                                       DAUX
      CALL ELTIME(2,9)                                                  DAUX
      RETURN                                                            DAUX
      END                                                               DAUX
```

```
      SUBROUTINE DAUX11                                                 DAUX11
C                                                REV IV    07/24/86SLIP
C     CALLED BY SUBROUTINE DAUX TO COMPUTE                              DAUX11
C                                                                       DAUX11
C                           -1                    -1                    DAUX11
C        (C11) = (B11)(M)   (A11) + (B12)(PHI)   (A21)                  DAUX11
C                                                                       DAUX11
C                          -1                   -1                      DAUX11
C        (R1) = (B11)(M)   (U1) + (B12)(PHI)   (U2) - (V1)              DAUX11
C                                                                       DAUX11
      IMPLICIT REAL*8(A-H,O-Z)                                          DAUX11
      COMMON/CONTRL/ TIME,NSEG,NJNT,NPL,NBLT,NBAG,NVEH,NGRND,           DAUX11
     *               NS,NQ,NSD,NFLX,NHRNSS,NWINDF,NJNTF,NPRT(36),NPG    PAGE
      COMMON/SGMNTS/ D(3,3,30),WMEG(3,30),WMEGD(3,30),U1(3,30),U2(3,30),DAUX11
     *               SEGLP(3,30),SEGLV(3,30),SEGLA(3,30),NSYM(30)       DAUX11
      COMMON/DESCRP/ PHI(3,30),W(30),RW(30),SR(4,60),HA(3,60),HB(3,60), SLIP
     *               RPHI(3,30),HT(3,3,60),SPRING(5,90),VISC(7,90),     DAUX11
     *               JNT(30),IPIN(30),ISING(30),IGLOB(30),JOINTF(30)    DAUX11
      COMMON/CMATRX/ V1(3,30),V2(3,30),V3(3,12),B12(3,3,60),A22(3,3,60),DAUX11
     *               F(3,30),TQ(3,30),WJ(30),A11(3,3,30)                SLIP
      COMMON/TEMPVS/ C(3,3,600),RHS(3,54),IJK(54,54),IJ,NQ2S            CHGIII
     *              ,DN(3,3),DM(3,3),SN(3,3),SM(3,3),HH(3,3),BN(3)      DAUX11
      CALL ELTIME(1,14)                                                 DAUX11
      DO 30 M=1,NJNT                                                    DAUX11
      N = IABS(JNT(M))                                                  DAUX11
      MQ = NQ2S + M                                                     DAUX11
      IJ = IJ+1                                                         DAUX11
      IJK(MQ,MQ) = IJ                                                   DAUX11
      IF (N.GT.0) GO TO 13                                              DAUX11
C                                                                       DAUX11
C     IF (N < 1)  SET  C11(M,M) = I                                     DAUX11
C                                                                       DAUX11
C                 AND  RHS(M)   = V1(M)                                 DAUX11
C                                                                       DAUX11
      DO 12 I=1,3                                                       DAUX11
      DO 11 J=1,3                                                       DAUX11
   11 C(I,J,IJ) = 0.0                                                   DAUX11
      C(I,I,IJ) = 1.0                                                   DAUX11
   12 RHS(I,MQ) = V1(I,M)                                               DAUX11
      IJK(MQ,MQ) = -IJ                                                  DAUX11
      GO TO 30                                                          DAUX11
C                                                                       DAUX11
C     IF (N > 0)  SET  RHS(M)   = U1(N) - U1(M+1) - V1(M)               DAUX11
C                               + B12(M,N)U2(N) + B12(M,M+1)U2(M+1)     DAUX11
C                                                                       DAUX11
C                 AND  C11(M,N) = RW(N) + RW(M+1)                       DAUX11
C                               + B12(M,N   )PHI(N   )'A21(N   ,M)      DAUX11
C                               + B12(M,M+1)PHI(M+1)'A21(M+1,M)         DAUX11
C                                                                       DAUX11
   13 DO 15 I=1,3                                                       DAUX11
```

```
      T1 = -V1(I,M)                                                     SLIP
      DO 15 J = 1,3                                                     DAUX11
      T1 = T1 + B12(I,J,2*M-1)*U2(J,N) + B12(I,J,2*M)*U2(J,M+1)        DAUX11
     *        + A11(I,J,M)*(U1(J,N) - U1(J,M+1))                        SLIP
      IF (J.LT.I) GO TO 15                                              DAUX11
      T2 = 0.0                                                          DAUX11
      IF (J.EQ.I) T2 = RW(N) + RW(M+1)                                  DAUX11
      DO 14 K=1,3                                                       DAUX11
   14 T2 = T2 + B12(I,K,2*M-1)*RPHI(K,N  )*B12(J,K,2*M-1)               DAUX11
     *        + B12(I,K,2*M  )*RPHI(K,M+1)*B12(J,K,2*M  )               DAUX11
      C(I,J,IJ) = T2                                                    DAUX11
      C(J,I,IJ) = T2                                                    DAUX11
   15 RHS(I,MQ) = T1                                                    DAUX11
      IF  (ISING(N).NE.0)  GO TO 30                                     DAUX11
      L = 0                                                             DAUX11
      IF (N.GT.1) L = IABS(JNT(N-1))                                    DAUX11
      IF (L.EQ.0) GO TO 18                                              DAUX11
C                                                                       DAUX11
C     IF (N > 1) AND (L = JNT(N-1) > 0)                                 DAUX11
C                                                                       DAUX11
C         SET  C11(M,N-1) =  -RW(N) + B12(M,N)PHI(N)'A21(N,N-1)         DAUX11
C                                                                       DAUX11
C                                    T                                  DAUX11
C         AND  C11(N-1,M) = C(M,N-1)                                    DAUX11
C                                                                       DAUX11
      KJNT = NQ2S + N -1                                                DAUX11
      IJ = IJ+1                                                         DAUX11
      IJK(MQ,KJNT) = IJ                                                 DAUX11
      IJK(KJNT,MQ) = IJ+1                                               DAUX11
      DO 17 I=1,3                                                       DAUX11
      DO 17 J=1,3                                                       DAUX11
      C(I,J,IJ) = 0.0                                                   DAUX11
      DO 16 K=1,3                                                       DAUX11
   16 C(I,J,IJ) = C(I,J,IJ) + B12(I,K,2*M-1)*RPHI(K,N)*B12(J,K,2*N-2)   DAUX11
     *                      - A11(I,K,M)*RW(N)*A11(J,K,N-1)             SLIP
   17 C(J,I,IJ+1) = C(I,J,IJ)                                           DAUX11
      IJ = IJ+1                                                         DAUX11
   18 IF (M.EQ.NJNT) GO TO 30                                           DAUX11
      M1 = M+1                                                          DAUX11
      DO 21 L=M1,NJNT                                                   DAUX11
      IF (IABS(JNT(L)).NE.N) GO TO 21                                   DAUX11
C                                                                       DAUX11
C     IF (L > M) AND (JNT(L) = N)                                       DAUX11
C                                                                       DAUX11
C         SET  C11(M,L) = RW(N) + B12(M,N)PHI(N)'A21(N,L)               DAUX11
C                                                                       DAUX11
C                                 T                                     DAUX11
C         AND  C11(L,M) = C11(M,L)                                      DAUX11
C                                                                       DAUX11
      KJNT = NQ2S + L                                                   DAUX11
```

```
      IJ = IJ+1                                                    DAUX11
      IJK(MQ,KJNT) = IJ                                            DAUX11
      IJK(KJNT,MQ) = IJ+1                                          DAUX11
      DO 20 I=1,3                                                  DAUX11
      DO 20 J=1,3                                                  DAUX11
      C(I,J,IJ) = 0.0                                              DAUX11
      DO 19 K=1,3                                                  DAUX11
   19 C(I,J,IJ) = C(I,J,IJ) + B12(I,K,2*M-1)*RPHI(K,N)*B12(J,K,2*L-1) DAUX11
     *                      + A11(I,K,M)*RW(N)*A11(J,K,L)          SLIP
   20 C(J,I,IJ+1) = C(I,J,IJ)                                      DAUX11
      IJ = IJ+1                                                    DAUX11
   21 CONTINUE                                                     DAUX11
   30 CONTINUE                                                     DAUX11
      CALL ELTIME(2,14)                                            DAUX11
      RETURN                                                       DAUX11
      END                                                          DAUX11
```

```
      SUBROUTINE DAUX12                                                 DAUX12
C                                                   REV IV    07/24/86SLIP
C     CALLED BY SUBROUTINE DAUX TO COMPUTE                              DAUX12
C                                                                       DAUX12
C                                -1                                     DAUX12
C        (C12) = (B12)(PHI)   (A22)                                     DAUX12
C                                                                       DAUX12
C                           T                                           DAUX12
C        (C21) = (C12)                                                  DAUX12
C                                                                       DAUX12
      IMPLICIT REAL*8(A-H,O-Z)                                          DAUX12
      COMMON/CONTRL/ TIME,NSEG,NJNT,NPL,NBLT,NBAG,NVEH,NGRND,           DAUX12
     *               NS,NQ,NSD,NFLX,NHRNSS,NWINDF,NJNTF,NPRT(36),NPG    PAGE
      COMMON/DESCRP/ PHI(3,30),W(30),RW(30),SR(4,60),HA(3,60),HB(3,60), SLIP
     *               RPHI(3,30),HT(3,3,60),SPRING(5,90),VISC(7,90),     DAUX12
     *               JNT(30),IPIN(30),ISING(30),IGLOB(30),JOINTF(30)    DAUX12
      COMMON/CMATRX/ V1(3,30),V2(3,30),V3(3,12),B12(3,3,60),A22(3,3,60),DAUX12
     *               F(3,30),TQ(3,30),WJ(30),A11(3,3,30)                SLIP
      LOGICAL*1 FREE                                                    SLIP
      COMMON/TEMPVS/ C(3,3,600),RHS(3,54),IJK(54,54),IJ,NQ2S            CHGIII
     *              ,DN(3,3),DM(3,3),SN(3,3),SM(3,3),HH(3,3),BN(3)      DAUX12
     *              ,IDUM(362),FREE(30)                                 SLIP
      CALL ELTIME(1,15)                                                 DAUX12
      NQSJNT = NQ2S + NJNT                                              DAUX12
      DO 60 M=1,NJNT                                                    DAUX12
      N = IABS(JNT(M))                                                  DAUX12
      IF (N.EQ.0) GO TO 60                                              DAUX12
      MQ = NQ2S + M                                                     DAUX12
      IF (FREE(M)) GO TO 37                                             SLIP
      MJNT = NQSJNT + M                                                 DAUX12
      IJ = IJ+1                                                         DAUX12
      IJK(MQ,MJNT) = IJ                                                 DAUX12
      IJK(MJNT,MQ) = IJ+1                                               DAUX12
      DO 36 I=1,3                                                       DAUX12
      DO 36 J=1,3                                                       DAUX12
      SN(I,J) = 0.0                                                     DAUX12
      SM(I,J) = 0.0                                                     DAUX12
      DO 35 K=1,3                                                       DAUX12
      SN(I,J) = SN(I,J) + B12(I,K,2*M-1) * RPHI(K,N  ) * A22(K,J,2*M-1) DAUX12
   35 SM(I,J) = SM(I,J) + B12(I,K,2*M  ) * RPHI(K,M+1) * A22(K,J,2*M  ) DAUX12
      C(I,J,IJ  ) = SN(I,J) - SM(I,J)                                   DAUX12
   36 C(J,I,IJ+1) = C(I,J,IJ)                                           DAUX12
      IJ = IJ+1                                                         DAUX12
   37 IF  (ISING(N).NE.0)  GO TO 50                                     DAUX12
      IF  (N.EQ.1)  GO TO 43                                            DAUX12
      IF (FREE(N-1)) GO TO 43                                           SLIP
      MJNT = NQSJNT + N-1                                               DAUX12
      IJ = IJ+1                                                         DAUX12
      IJK(MQ,MJNT) = IJ                                                 DAUX12
      IJK(MJNT,MQ) = IJ+1                                               DAUX12
```

```
      DO 42 I=1,3                                                      DAUX12
      DO 42 J=1,3                                                      DAUX12
      SN(I,J) = 0.0                                                    DAUX12
      DO 41 K=1,3                                                      DAUX12
   41 SN(I,J) = SN(I,J) + B12(I,K,2*M-1) * RPHI(K,N  ) * A22(K,J,2*N-2) DAUX12
      C(I,J,IJ  ) = -SN(I,J)                                           DAUX12
   42 C(J,I,IJ+1) = -SN(I,J)                                           DAUX12
      IJ = IJ+1                                                        DAUX12
   43 DO 49 L=N,NJNT                                                   DAUX12
      IF (L.EQ.M) GO TO 49                                             DAUX12
      IF (IABS(JNT(L)).NE.N ) GO TO 49                                 DAUX12
      IF (FREE(L)) GO TO 49                                            SLIP
      MJNT = NQSJNT + L                                                DAUX12
      IJ = IJ+1                                                        DAUX12
      IJK(MQ,MJNT) = IJ                                                DAUX12
      IJK(MJNT,MQ) = IJ+1                                              DAUX12
      DO 48 I=1,3                                                      DAUX12
      DO 48 J=1,3                                                      DAUX12
      SN(I,J) = 0.0                                                    DAUX12
      DO 47 K=1,3                                                      DAUX12
   47 SN(I,J) = SN(I,J) + B12(I,K,2*M-1) * RPHI(K,N  ) * A22(K,J,2*L-1) DAUX12
      C(I,J,IJ  ) = SN(I,J)                                            DAUX12
   48 C(J,I,IJ+1) = SN(I,J)                                            DAUX12
      IJ = IJ +1                                                       DAUX12
   49 CONTINUE                                                         DAUX12
   50 IF  (M.EQ.NJNT)  GO TO 60                                        DAUX12
      IF  (ISING(M+1).NE.0)  GO TO 60                                  DAUX12
      M1 = M+1                                                         DAUX12
      DO 59 L=M1,NJNT                                                  DAUX12
      IF (IABS(JNT(L)).NE.M1) GO TO 59                                 DAUX12
      IF (FREE(L)) GO TO 59                                            SLIP
      MJNT = NQSJNT + L                                                DAUX12
      IJ = IJ+1                                                        DAUX12
      IJK(MQ,MJNT) = IJ                                                DAUX12
      IJK(MJNT,MQ) = IJ+1                                              DAUX12
      DO 58 I=1,3                                                      DAUX12
      DO 58 J=1,3                                                      DAUX12
      SM(I,J) = 0.0                                                    DAUX12
      DO 57 K=1,3                                                      DAUX12
   57 SM(I,J) = SM(I,J) + B12(I,K,2*M  ) * RPHI(K,M+1) * A22(K,J,2*L-1) DAUX12
      C(I,J,IJ  ) = SM(I,J)                                            DAUX12
   58 C(J,I,IJ+1) = SM(I,J)                                            DAUX12
      IJ = IJ +1                                                       DAUX12
   59 CONTINUE                                                         DAUX12
   60 CONTINUE                                                         DAUX12
      CALL ELTIME(2,15)                                                DAUX12
      RETURN                                                           DAUX12
      END                                                              DAUX12
```

```
      SUBROUTINE DAUX22                                                 DAUX22
C                                               REV IV    07/24/86SLIP
C     CALLED BY SUBROUTINE DAUX TO COMPUTE                              DAUX22
C                                                                       DAUX22
C                                                                       DAUX22
C                               -1                                      DAUX22
C        (C22) = (B22)(PHI)   (A22) - (B24)                             DAUX22
C                                                                       DAUX22
C                               -1                                      DAUX22
C        (R2)  = (B22)(PHI)   (U2)  - (V2)                              DAUX22
C                                                                       DAUX22
      IMPLICIT REAL*8(A-H,O-Z)                                          DAUX22
      COMMON/CONTRL/ TIME,NSEG,NJNT,NPL,NBLT,NBAG,NVEH,NGRND,           DAUX22
     *               NS,NQ,NSD,NFLX,NHRNSS,NWINDF,NJNTF,NPRT(36),NPG    PAGE
      COMMON/SGMNTS/ D(3,3,30),WMEG(3,30),WMEGD(3,30),U1(3,30),U2(3,30),DAUX22
     *               SEGLP(3,30),SEGLV(3,30),SEGLA(3,30),MSYM(30)       DAUX22
      COMMON/DESCRP/ PHI(3,30),W(30),RW(30),SR(4,60),HA(3,60),HB(3,60), SLIP
     *               RPHI(3,30),HT(3,3,60),SPRING(5,90),VISC(7,90),     DAUX22
     *               JNT(30),IPIN(30),ISING(30),IGLOB(30),JOINTF(30)    DAUX22
      COMMON/CMATRX/ V1(3,30),V2(3,30),V3(3,12),B12(3,3,60),A22(3,3,60),DAUX22
     *               F(3,30),TQ(3,30),WJ(30),A11(3,3,30)                SLIP
      COMMON/CEULER/ IEULER(30),HIR(3,3,90),ANG(3,30),ANGD(3,30),       JDRIFT
     *               FE(3,30),TQE(3,30),CONST(5,30)                     JDRIFT
      LOGICAL*1 FREE                                                    SLIP
      COMMON/TEMPVS/ C(3,3,600),RHS(3,54),IJK(54,54),IJ,NQ2S            CHGIII
     *              ,DN(3,3),DM(3,3),SN(3,3),SM(3,3),HH(3,3),BN(3)      DAUX22
     *              ,IDUM(362),FREE(30)                                 SLIP
      LOGICAL TEST                                                      DAUX22
      CALL ELTIME(1,16)                                                 DAUX22
      NQSJNT = NQ2S + NJNT                                              DAUX22
      DO 90 M=1,NJNT                                                    DAUX22
      MJNT = NQSJNT + M                                                 DAUX22
      DO 60 I=1,3                                                       DAUX22
   60 RHS(I,MJNT) = V2(I,M)                                             DAUX22
      N = IABS(JNT(M))                                                  DAUX22
      IF (N.EQ.0) GO TO 90                                              DAUX22
      IF (FREE(M)) GO TO 90                                             SLIP
      IJ = IJ+1                                                         DAUX22
      IJK(MJNT,MJNT) = IJ                                               DAUX22
      DO 61 J=1,3                                                       DAUX22
      DO 61 I=1,3                                                       DAUX22
   61 HH(I,J) = 0.0                                                     DAUX22
      LGO = IPIN(M)+8                                                   SLIP
      TEST = .FALSE.                                                    DAUX22
      GO TO (64,64,64,62,64,64,64,64,63,64,64,64,64,63,63),LGO          SLIP
   62 IF (IEULER(M).GE.7) GO TO 64                                      DAUX22
      TEST = IEULER(M).LT.4                                             DAUX22
   63 AN = 0.0                                                          DAUX22
      DO 51 J=1,3                                                       DAUX22
   51 AN = AN + HB(J,2*M-1)**2 * RPHI(J,N )                             DAUX22
```

```
     *        + HB(J,2*M  )**2 * RPHI(J,M+1)                          DAUX22
      IF (TEST) GO TO 64                                              DAUX22
      CALL DOT31 (D(1,1,N),HB(1,2*M-1),BN)                            DAUX22
      DO 53 J=1,3                                                     DAUX22
      DO 53 I=1,3                                                     DAUX22
   53 HH(I,J) = AN*BN(I)*BN(J)                                        DAUX22
   64 DO 67 I=1,3                                                     DAUX22
      RHS(I,MJNT) = -V2(I,M)                                          DAUX22
      DO 66 J=1,3                                                     DAUX22
      RHS(I,MJNT) = RHS(I,MJNT) + A22(J,I,2*M-1)*U2(J,N  )            DAUX22
     *                          - A22(J,I,2*M  )*U2(J,M+1)            DAUX22
      SN(I,J) = 0.0                                                   DAUX22
      IF (TEST) GO TO 66                                              DAUX22
      DO 65 K=1,3                                                     DAUX22
   65 SN(I,J) = SN(I,J) + A22(K,I,2*M-1) * RPHI(K,N  ) * A22(K,J,2*M-1) DAUX22
     *                  + A22(K,I,2*M  ) * RPHI(K,M+1) * A22(K,J,2*M  ) DAUX22
   66 C(I,J,IJ) = SN(I,J) + HH(I,J)                                   DAUX22
   67 IF (TEST) C(I,I,IJ) = AN                                        DAUX22
      IF  (ISING(N).NE.0)  GO TO 90                                   DAUX22
      IF (N.EQ.1) GO TO 80                                            DAUX22
      IF (FREE(N-1)) GO TO 80                                         SLIP
      N1JNT = NQSJNT + N -1                                           DAUX22
      IJ = IJ+1                                                       DAUX22
      IJK(MJNT,N1JNT) = IJ                                            DAUX22
      IJK(N1JNT,MJNT) = IJ+1                                          DAUX22
      DO 77 I=1,3                                                     DAUX22
      DO 77 J=1,3                                                     DAUX22
      SN(I,J) = 0.0                                                   DAUX22
      DO 76 K=1,3                                                     DAUX22
   76 SN(I,J) = SN(I,J) + A22(K,I,2*M-1) * RPHI(K,N  ) * A22(K,J,2*N-2) DAUX22
      C(I,J,IJ)   = -SN(I,J)                                          DAUX22
   77 C(J,I,IJ+1) = -SN(I,J)                                          DAUX22
      IJ = IJ+1                                                       DAUX22
   80 IF (M.EQ.NJNT) GO TO 90                                         DAUX22
      M1 = M+1                                                        DAUX22
      DO  88 L=M1,NJNT                                                DAUX22
      IF (IABS(JNT(L)).NE.N) GO TO 88                                 DAUX22
      IF (FREE(L)) GO TO 88                                           SLIP
      LJNT = NQSJNT + L                                               DAUX22
      IJ = IJ+1                                                       DAUX22
      IJK(MJNT,LJNT) = IJ                                             DAUX22
      IJK(LJNT,MJNT) = IJ+1                                           DAUX22
      DO 87 I=1,3                                                     DAUX22
      DO 87 J=1,3                                                     DAUX22
      SN(I,J) = 0.0                                                   DAUX22
      DO 86 K=1,3                                                     DAUX22
   86 SN(I,J) = SN(I,J) + A22(K,I,2*M-1) * RPHI(K,N  ) * A22(K,J,2*L-1) DAUX22
      C(I,J,IJ)   = SN(I,J)                                           DAUX22
   87 C(J,I,IJ+1) = SN(I,J)                                           DAUX22
      IJ = IJ+1                                                       DAUX22
```

```
   88 CONTINUE                                                        DAUX22
   90 CONTINUE                                                        DAUX22
      CALL ELTIME(2,16)                                               DAUX22
      RETURN                                                          DAUX22
      END                                                             DAUX22
```

```
      SUBROUTINE DAUX31                                                 DAUX31
C                                                   REV IV    07/24/86SLIP
C     CALLED BY SUBROUTINE DAUX TO COMPUTE                              DAUX31
C                                                                       DAUX31
C                     -1                   -1                           DAUX31
C     (C13) = (B11)(M)   (A13) + (B12)(PHI)   (A23)                     DAUX31
C                                                                       DAUX31
C                     -1                   -1                           DAUX31
C     (C31) = (B31)(M)   (A11) + (B32)(PHI)   (A21)                     DAUX31
C                                                                       DAUX31
      IMPLICIT REAL*8 (A-H,O-Z)                                         DAUX31
      COMMON/CONTRL/ TIME,NSEG,NJNT,NPL,NBLT,NBAG,NVEH,NGRND,           DAUX31
     *               NS,NQ,NSD,NFLX,NHRNSS,NWINDF,NJNTF,NPRT(36),NPG    PAGE
      COMMON/DESCRP/ PHI(3,30),W(30),RW(30),SR(4,60),HA(3,60),HB(3,60), SLIP
     *               RPHI(3,30),HT(3,3,60),SPRING(5,90),VISC(7,90),     DAUX31
     *               JNT(30),IPIN(30),ISING(30),IGLOB(30),JOINTF(30)    DAUX31
      COMMON/CMATRX/ V1(3,30),V2(3,30),V3(3,12),B12(3,3,60),A22(3,3,60),DAUX31
     *               F(3,30),TQ(3,30),WJ(30),A11(3,3,30)                SLIP
      COMMON/CSTRNT/ A13(3,3,24),A23(3,3,24),B31(3,3,24),B32(3,3,24),   DAUX31
     *               HHT(3,3,12),RK1(3,12),RK2(3,12),QQ(3,12),TQQ(3,12),DAUX31
     *               RQQ(3,12),HQQ(3,12),SQQ(12),CFQQ(12),              DAUX31
     *               KQ1(12),KQ2(12),KQTYPE(12)                         DAUX31
      COMMON/TEMPVS/ C(3,3,600),RHS(3,54),IJK(54,54),IJ,NQ2S            CHGIII
      CALL ELTIME(1,17)                                                 DAUX31
      DO 30 N=1,NQ                                                      DAUX31
      IF (KQTYPE(N).LT.0) GO TO 30                                      DAUX31
      K1 = KQ1(N)                                                       DAUX31
      K2 = KQ2(N)                                                       DAUX31
      NNS = NQ2S - NQ + N                                               DAUX31
      IF (K1.LE.1) GO TO 13                                             DAUX31
      IF (IABS(JNT(K1-1)).EQ.0) GO TO 13                                DAUX31
      IF  (ISING(K1).NE.0)  GO TO 13                                    DAUX31
C                                                                       DAUX31
C                                    -1                                 DAUX31
C     C13(K1-1,N) = B11(K1-1,K1)M      (K1)A13(K1,N)                    DAUX31
C                                      -1                               DAUX31
C                 + B12(K1-1,K1)PHI   (K1)A23(K1,N)                     DAUX31
C                                                                       DAUX31
C                                 -1                                    DAUX31
C     C31(N,K1-1) = B31(N,K1)M      (K1)A11(K1,K1-1)                    DAUX31
C                                   -1                                  DAUX31
C                 + B32(N,K1)PHI   (K1)A21(K1,K1-1)                     DAUX31
C                                                                       DAUX31
      MQ = NQ2S + K1 - 1                                                DAUX31
      IJ = IJ+1                                                         DAUX31
      IJK(MQ,NNS) = IJ                                                  DAUX31
      IJK(NNS,MQ) = IJ+1                                                DAUX31
      DO 12 I=1,3                                                       DAUX31
      DO 12 J=1,3                                                       DAUX31
      SUM = 0.0                                                         SLIP
```

```
      TUM = 0.0                                                         SLIP
      DO 11 K=1,3                                                       DAUX31
      SUM = SUM + B12(I,K,2*K1-2)*RPHI(K,K1)*A23(K,J,2*N-1 )            DAUX31
     *          - A11(I,K,K1-1)*RW(K1)*A13(K,J,2*N-1)                   SLIP
   11 TUM = TUM + B32(I,K,2*N-1 )*RPHI(K,K1)*B12(J,K,2*K1-2)            DAUX31
     *          - B31(I,K,2*N-1)*RW(K1)*A11(K,J,K1-1)                   SLIP
      C(I,J,IJ)    = SUM                                                DAUX31
   12 C(I,J,IJ+1) = TUM                                                 DAUX31
      IJ = IJ+1                                                         DAUX31
   13 IF (K2.LE.1) GO TO 16                                             DAUX31
      IF (IABS(JNT(K2-1)).EQ.0) GO TO 16                                DAUX31
      IF  (ISING(K2).NE.0)  GO TO 16                                    DAUX31
C                                                                       DAUX31
C                                    -1                                 DAUX31
C     C13(K2-1,N) = B11(K2-1,K2)M      (K2)A13(K2,N)                    DAUX31
C                                       -1                              DAUX31
C                 + B12(K2-1,K2)PHI   (K2)A23(K2,N)                     DAUX31
C                                                                       DAUX31
C                                -1                                     DAUX31
C     C31(N,K2-1) = B31(N,K2)M      (K2)A11(K2,K2-1)                    DAUX31
C                                   -1                                  DAUX31
C                 + B32(N,K2)PHI   (K2)A21(K2,K2-1)                     DAUX31
C                                                                       DAUX31
      MQ = NQ2S + K2 - 1                                                DAUX31
      IJ = IJ+1                                                         DAUX31
      IJK(MQ,NNS) = IJ                                                  DAUX31
      IJK(NNS,MQ) = IJ+1                                                DAUX31
      DO 15 I=1,3                                                       DAUX31
      DO 15 J=1,3                                                       DAUX31
      SUM = 0.0                                                         SLIP
      TUM = 0.0                                                         SLIP
      DO 14 K=1,3                                                       DAUX31
      SUM = SUM + B12(I,K,2*K2-2)*RPHI(K,K2)*A23(K,J,2*N    )           DAUX31
     *          - A11(I,K,K2-1)*RW(K2)*A13(K,J,2*N)                     SLIP
   14 TUM = TUM + B32(I,K,2*N    )*RPHI(K,K2)*B12(J,K,2*K2-2)           DAUX31
     *          - B31(I,K,2*N)*RW(K2)*A11(K,J,K2-1)                     SLIP
      C(I,J,IJ)   = SUM                                                 DAUX31
   15 C(I,J,IJ+1) = TUM                                                 DAUX31
      IJ = IJ+1                                                         DAUX31
   16 IF (NJNT.LE.0) GO TO 30                                           DAUX31
      DO 26 L=1,NJNT                                                    DAUX31
      IF (IABS(JNT(L)).NE.K1) GO TO 21                                  DAUX31
      IF  (ISING(K1).NE.0)  GO TO 21                                    DAUX31
C                                                                       DAUX31
C     FOR ANY L SUCH THAT JNT(L) = K1                                   DAUX31
C                                                                       DAUX31
C                             -1                                        DAUX31
C     C13(L,N) = B11(L,K1)M      (K1)A13(K1,N)                          DAUX31
C                                -1                                     DAUX31
C              + B12(L,K1)PHI   (K1)A23(K1,N)                           DAUX31
```

```
C                                                                    DAUX31
C                               -1                                   DAUX31
C     C31(N,L) = B31(N,K1)M     (K1)A11(K1,L)                        DAUX31
C                                 -1                                 DAUX31
C              + B32(N,K1)PHI   (K1)A21(K1,L)                        DAUX31
C                                                                    DAUX31
      MQ = NQ2S + L                                                  DAUX31
      IF (IJK(MQ,NNS).NE.0) GO TO 18                                 DAUX31
      IJ = IJ+1                                                      DAUX31
      IJK(MQ,NNS) = IJ                                               DAUX31
      IJK(NNS,MQ) = IJ+1                                             DAUX31
      DO 17 J=1,3                                                    DAUX31
      DO 17 I=1,3                                                    DAUX31
      C(I,J,IJ  ) = 0.0                                              DAUX31
   17 C(I,J,IJ+1) = 0.0                                              DAUX31
      IJ = IJ+1                                                      DAUX31
   18 JJ = IJK(MQ,NNS)                                               DAUX31
      DO 20 I=1,3                                                    DAUX31
      DO 20 J=1,3                                                    DAUX31
      SUM = C(I,J,JJ)                                                SLIP
      TUM = C(I,J,JJ+1)                                              SLIP
      DO 19 K=1,3                                                    DAUX31
      SUM = SUM + B12(I,K,2*L-1)*RPHI(K,K1)*A23(K,J,2*N-1)           DAUX31
     *          +A11(I,K,L)*RW(K1)*A13(K,J,2*N-1)                    SLIP
   19 TUM = TUM + B32(I,K,2*N-1)*RPHI(K,K1)*B12(J,K,2*L-1)           DAUX31
     *          +B31(I,K,2*N-1)*RW(K1)*A11(J,K,L)                    SLIP
      C(I,J,JJ)   = SUM                                              DAUX31
   20 C(I,J,JJ+1) = TUM                                              DAUX31
   21 IF (IABS(JNT(L)).NE.K2) GO TO 26                               DAUX31
      IF  (ISING(K2).NE.0)  GO TO 26                                 DAUX31
C                                                                    DAUX31
C     FOR ANY L SUCH THAT JNT(L) = K2                                DAUX31
C                                                                    DAUX31
C                               -1                                   DAUX31
C     C13(L,N) = B11(L,K2)M     (K2)A13(K2,N)                        DAUX31
C                                 -1                                 DAUX31
C              + B12(L,K2)PHI   (K2)A23(K2,N)                        DAUX31
C                                                                    DAUX31
C                               -1                                   DAUX31
C     C31(N,L) = B31(N,K2)M     (K2)A11(K2,L)                        DAUX31
C                                 -1                                 DAUX31
C              + B32(N,K2)PHI   (K2)A21(K2,L)                        DAUX31
C                                                                    DAUX31
      MQ = NQ2S + L                                                  DAUX31
      IF (IJK(MQ,NNS).NE.0) GO TO 23                                 DAUX31
      IJ = IJ+1                                                      DAUX31
      IJK(MQ,NNS) = IJ                                               DAJX31
      IJK(NNS,MQ) = IJ+1                                             DAUX31
      DO 22 J=1,3                                                    DAUX31
      DO 22 I=1,3                                                    DAUX31
```

```
      C(I,J,IJ  ) = 0.0                                             DAUX31
   22 C(I,J,IJ+1) = 0.0                                             DAUX31
      IJ = IJ+1                                                     DAUX31
   23 JJ = IJK(MQ,MNS)                                              DAUX31
      DO 25 I=1,3                                                   DAUX31
      DO 25 J=1,3                                                   DAUX31
      SUM = C(I,J,JJ)                                               SLIP
      TUM = C(I,J,JJ+1)                                             SLIP
      DO 24 K=1,3                                                   DAUX31
      SUM = SUM + B12(I,K,2*L-1)*RPHI(K,K2)*A23(K,J,2*N  )          DAUX31
     *          + A11(I,K,L)*RW(K2)*A13(K,J,2*N)                    SLIP
   24 TUM = TUM + B32(I,K,2*N  )*RPHI(K,K2)*B12(J,K,2*L-1)          DAUX31
     *          + B31(I,K,2*N)*RW(K2)*A11(J,K,L)                    SLIP
      C(I,J,JJ)   = SUM                                             DAUX31
   25 C(I,J,JJ+1) = TUM                                             DAUX31
   26 CONTINUE                                                      DAUX31
   30 CONTINUE                                                      DAUX31
      CALL ELTIME(2,17)                                             DAUX31
      RETURN                                                        DAUX31
      END                                                           DAUX31
```

```
      SUBROUTINE DAUX32                                                 DAUX32
C                                                  REV IV    07/24/86SLIP
C     CALLED BY SUBROUTINE DAUX TO COMPUTE                              DAUX32
C                                                                       DAUX32
C                             -1                                        DAUX32
C     (C23) = (B22)(PHI)   (A23)                                        DAUX32
C                                                                       DAUX32
C                             -1                                        DAUX32
C     (C32) = (B32)(PHI)   (A22)                                        DAUX32
C                                                                       DAUX32
      IMPLICIT REAL*8 (A-H,O-Z)                                         DAUX32
      COMMON/CONTRL/ TIME,NSEG,NJNT,NPL,NBLT,NBAG,NVEH,NGRND,           DAUX32
     *               NS,NQ,NSD,NFLX,NHRNSS,NWINDF,NJNTF,NPRT(36),NPG    PAGE
      COMMON/DESCRP/ PHI(3,30),W(30),RW(30),SR(4,60),HA(3,60),HB(3,60), SLIP
     *               RPHI(3,30),HT(3,3,60),SPRING(5,90),VISC(7,90),     DAUX32
     *               JNT(30),IPIN(30),ISING(30),IGLOB(30),JOINTF(30)    DAUX32
      COMMON/CMATRX/ V1(3,30),V2(3,30),V3(3,12),B12(3,3,60),A22(3,3,60),DAUX32
     *               F(3,30),TQ(3,30),WJ(30),A11(3,3,30)                SLIP
      COMMON/CSTRNT/ A13(3,3,24),A23(3,3,24),B31(3,3,24),B32(3,3,24),   DAUX32
     *               HHT(3,3,12),RK1(3,12),RK2(3,12),QQ(3,12),TQQ(3,12),DAUX32
     *               RQQ(3,12),HQQ(3,12),SQQ(12),CFQQ(12),              DAUX32
     *               KQ1(12),KQ2(12),KQTYPE(12)                         DAUX32
      LOGICAL*1 FREE                                                    SLIP
      COMMON/TEMPVS/ C(3,3,600),RHS(3,54),IJK(54,54),IJ,NQ2S            CHGIII
     *              ,DN(3,3),DM(3,3),BN(3),IDUM(416),FREE(30)           SLIP
      CALL ELTIME(1,16)                                                 DAUX32
      NQSJNT = NQ2S + NJNT                                              DAUX32
      DO 60 N=1,NQ                                                      DAUX32
      IF (KQTYPE(N).LT.0) GO TO 60                                      DAUX32
      K1 = KQ1(N)                                                       DAUX32
      K2 = KQ2(N)                                                       DAUX32
      NNS = NQ2S - NQ + N                                               DAUX32
      IF (K1.LE.1)        GO TO 43                                      DAUX32
      IF (IABS(JNT(K1-1)).EQ.0) GO TO 43                                DAUX32
      IF (FREE(K1-1)) GO TO 43                                          SLIP
      IF  (ISING(K1).NE.0)  GO TO 43                                    DAUX32
C                                                                       DAUX32
C                                        -1                             DAUX32
C     C23(K1-1,N) = B22(K1-1,K1)PHI   (K1)A23(K1,N)                     DAUX32
C                                                                       DAUX32
C                                     -1                                DAUX32
C     C32(N,K1-1) = B32(N,K1)PHI   (K1)A22(K1,K1-1)                     DAUX32
C                                                                       DAUX32
      KJNT = NQSJNT + K1 - 1                                            DAUX32
      IJ = IJ+1                                                         DAUX32
      IJK(KJNT,NNS) = IJ                                                DAUX32
      IJK(NNS,KJNT) = IJ+1                                              DAUX32
      DO 42 I=1,3                                                       DAUX32
      DO 42 J=1,3                                                       DAUX32
      SUM = 0.0                                                         DAUX32
```

```
      TUM = 0.0                                                         DAUX32
      DO 41 K=1,3                                                       DAUX32
      SUM = SUM + A22(K,I,2*K1-2) * RPHI(K,K1) * A23(K,J,2*N-1 )        DAUX32
   41 TUM = TUM + B32(I,K,2*N-1 ) * RPHI(K,K1) * A22(K,J,2*K1-2)        DAUX32
      C(I,J,IJ  ) = -SUM                                                DAUX32
   42 C(I,J,IJ+1) = -TUM                                                DAUX32
      IJ = IJ+1                                                         DAUX32
   43 IF (K2.LE.1)          GO TO 46                                    DAUX32
      IF (IABS(JNT(K2-1)).EQ.0) GO TO 46                                DAUX32
      IF (FREE(K2-1)) GO TO 46                                          SLIP
      IF  (ISING(K2).NE.0)  GO TO 46                                    DAUX32
C                                                                       DAUX32
C                                      -1                               DAUX32
C     C23(K2-1,N) = B22(K2-1,K2)PHI   (K2)A23(K2,N)                     DAUX32
C                                                                       DAUX32
C                                   -1                                  DAUX32
C     C32(N,K2-1) = B32(N,K2)PHI   (K2)A22(K2,K2-1)                     DAUX32
C                                                                       DAUX32
      KJNT = NQSJNT + K2 - 1                                            DAUX32
      IJ = IJ+1                                                         DAUX32
      IJK(KJNT,NNS) = IJ                                                DAUX32
      IJK(NNS,KJNT) = IJ+1                                              DAUX32
      DO 45 I=1,3                                                       DAUX32
      DO 45 J=1,3                                                       DAUX32
      SUM = 0.0                                                         DAUX32
      TUM = 0.0                                                         DAUX32
      DO 44 K=1,3                                                       DAUX32
      SUM = SUM + A22(K,I,2*K2-2) * RPHI(K,K2) * A23(K,J,2*N    )       DAUX32
   44 TUM = TUM + B32(I,K,2*N    ) * RPHI(K,K2) * A22(K,J,2*K2-2)       DAUX32
      C(I,J,IJ  ) = -SUM                                                DAUX32
   45 C(I,J,IJ+1) = -TUM                                                DAUX32
      IJ = IJ+1                                                         DAUX32
   46 IF (NJNT.LE.0)        GO TO 60                                    DAUX32
      DO 56 L=1,NJNT                                                    DAUX32
      IF (FREE(L)) GO TO 56                                             SLIP
      IF (IABS(JNT(L)).NE.K1) GO TO 51                                  DAUX32
      IF  (ISING(K1).NE.0)  GO TO 51                                    DAUX32
C                                                                       DAUX32
C     FOR ANY L SUCH THAT JNT(L) = K1                                   DAUX32
C                                                                       DAUX32
C                                -1                                     DAUX32
C     C23(L,N) = B22(L,K1)PHI   (K1)A23(K1,N)                           DAUX32
C                                                                       DAUX32
C                                -1                                     DAUX32
C     C32(N,L) = B32(N,K1)PHI   (K1)A22(K1,L)                           DAUX32
C                                                                       DAUX32
      KJNT = NQSJNT + L                                                 DAUX32
      IF (IJK(KJNT,NNS).NE.0) GO TO 48                                  DAUX32
      IJ = IJ+1                                                         DAUX32
      IJK(KJNT,NNS) = IJ                                                DAUX32
```

```
      IJK(NNS,KJNT) = IJ+1                                              DAUX32
      DO 47 J=1,3                                                       DAUX32
      DO 47 I=1,3                                                       DAUX32
      C(I,J,IJ  ) = 0.0                                                 DAUX32
   47 C(I,J,IJ+1) = 0.0                                                 DAUX32
      IJ = IJ+1                                                         DAUX32
   48 JJ = IJK(KJNT,NNS)                                                DAUX32
      DO 50 I=1,3                                                       DAUX32
      DO 50 J=1,3                                                       DAUX32
      SUM = C(I,J,JJ)                                                   DAUX32
      TUM = C(I,J,JJ+1)                                                 DAUX32
      DO 49 K=1,3                                                       DAUX32
      SUM = SUM + A22(K,I,2*L-1 ) * RPHI(K,K1) * A23(K,J,2*N-1 )        DAUX32
   49 TUM = TUM + B32(I,K,2*N-1 ) * RPHI(K,K1) * A22(K,J,2*L-1 )        DAUX32
      C(I,J,JJ)   = SUM                                                 DAUX32
   50 C(I,J,JJ+1) = TUM                                                 DAUX32
   51 IF (IABS(JNT(L)).NE.K2) GO TO 56                                  DAUX32
      IF  (ISING(K2).NE.0)  GO TO 56                                    DAUX32
C                                                                       DAUX32
C     FOR ANY L SUCH THAT JNT(L) = K2                                   DAUX32
C                                                                       DAUX32
C                                     -1                                DAUX32
C     C23(L,N) = B22(L,K2)PHI   (K2)A23(K2,N)                           DAUX32
C                                                                       DAUX32
C                                     -1                                DAUX32
C     C32(N,L) = B32(N,K2)PHI   (K2)A22(K2,L)                           DAUX32
C                                                                       DAUX32
      KJNT = NQSJNT + L                                                 DAUX32
      IF (IJK(KJNT,NNS).NE.0) GO TO 53                                  DAUX32
      IJ = IJ+1                                                         DAUX32
      IJK(KJNT,NNS) = IJ                                                DAUX32
      IJK(NNS,KJNT) = IJ+1                                              DAUX32
      DO 52 J=1,3                                                       DAUX32
      DO 52 I=1,3                                                       DAUX32
      C(I,J,IJ  ) = 0.0                                                 DAUX32
   52 C(I,J,IJ+1) = 0.0                                                 DAUX32
      IJ = IJ+1                                                         DAUX32
   53 JJ = IJK(KJNT,NNS)                                                DAUX32
      DO 55 I=1,3                                                       DAUX32
      DO 55 J=1,3                                                       DAUX32
      SUM = C(I,J,JJ)                                                   DAUX32
      TUM = C(I,J,JJ+1)                                                 DAUX32
      DO 54 K=1,3                                                       DAUX32
      SUM = SUM + A22(K,I,2*L-1 ) * RPHI(K,K2) * A23(K,J,2*N   )        DAUX32
   54 TUM = TUM + B32(I,K,2*N   ) * RPHI(K,K2) * A22(K,J,2*L-1 )        DAUX32
      C(I,J,JJ)   = SUM                                                 DAUX32
   55 C(I,J,JJ+1) = TUM                                                 DAUX32
   56 CONTINUE                                                          DAUX32
   60 CONTINUE                                                          DAUX32
      CALL ELTIME(2,18)                                                 DAUX32
```

```
      RETURN                                                          DAUX32
      END                                                             DAUX32
```

```
      SUBROUTINE DAUX33                                                 DAUX33
C                                                  REV IV    07/24/86SLIP
C     CALLED BY SUBROUTINE DAUX TO COMPUTE                              DAUX33
C                                                                       DAUX33
C                          -1                   -1                      DAUX33
C     (C33) = (B31)(M)   (A13) + (B32)(PHI)   (A23) - (B35)             DAUX33
C                                                                       DAUX33
C                          -1                   -1                      DAUX33
C     (R3)  = (B31)(M)   (U1)  + (B32)(PHI)   (U2)   - (V3)             DAUX33
C                                                                       DAUX33
      IMPLICIT REAL*8 (A-H,O-Z)                                         DAUX33
      COMMON/CONTRL/ TIME,NSEG,NJNT,NPL,NBLT,NBAG,NVEH,NGRND,           DAUX33
     *               NS,NQ,NSD,NFLX,NHRNSS,NWINDF,NJNTF,NPRT(36),NPG    PAGE
      COMMON/SGMNTS/ D(3,3,30),WMEG(3,30),WMEGD(3,30),U1(3,30),U2(3,30),DAUX33
     *               SEGLP(3,30),SEGLV(3,30),SEGLA(3,30),NSYM(30)       DAUX33
      COMMON/DESCRP/ PHI(3,30),W(30),RW(30),SR(4,60),HA(3,60),HB(3,60), SLIP
     *               RPHI(3,30),HT(3,3,60),SPRING(5,90),VISC(7,90),     DAUX33
     *               JNT(30),IPIN(30),ISING(30),IGLOB(30),JOINTF(30)    DAUX33
      COMMON/CMATRX/ V1(3,30),V2(3,30),V3(3,12),B12(3,3,60),A22(3,3,60),DAUX33
     *               F(3,30),TQ(3,30),WJ(30),A11(3,3,30)                SLIP
      COMMON/CSTRNT/ A13(3,3,24),A23(3,3,24),B31(3,3,24),B32(3,3,24),   DAUX33
     *               HHT(3,3,12),RK1(3,12),RK2(3,12),QQ(3,12),TQQ(3,12),DAUX33
     *               RQQ(3,12),HQQ(3,12),SQQ(12),CFQQ(12),              DAUX33
     *               KQ1(12),KQ2(12),KQTYPE(12)                         DAUX33
      COMMON/TEMPVS/ C(3,3,600),RHS(3,54),IJK(54,54),IJ,NQ2S            CHGIII
      CALL ELTIME(1,19)                                                 DAUX33
      DO 90 N=1,NQ                                                      DAUX33
      IF (KQTYPE(N).LT.0) GO TO 90                                      DAUX33
      K1 = KQ1(N)                                                       DAUX33
      K2 = KQ2(N)                                                       DAUX33
      NNS = NQ2S - NQ + N                                               DAUX33
C                                                                       DAUX33
C                            -1                        -1               DAUX33
C     RHS(N) = B31(N,K1)M   (K1)U1(K1) + B32(N,K1)PHI   (K1)U2(K1)      DAUX33
C                            -1                        -1               DAUX33
C            + B31(N,K2)M   (K2)U1(K2) + B32(N,K2)PHI   (K2)U2(K2)      DAUX33
C                                                                       DAUX33
C            - V3(N)                                                    DAUX33
C                                                                       DAUX33
      DO 63 I=1,3                                                       DAUX33
      SUM = 0.0                                                         DAUX33
      DO 62 K=1,3                                                       DAUX33
   62 SUM = SUM + B31(I,K,2*N-1)*U1(K,K1) + B32(I,K,2*N-1)*U2(K,K1)     DAUX33
     *          + B31(I,K,2*N  )*U1(K,K2) + B32(I,K,2*N  )*U2(K,K2)     DAUX33
   63 RHS(I,NNS) = SUM - V3(I,N)                                        DAUX33
C                                                                       DAUX33
C                              -1                        -1             DAUX33
C     C33(N,N) = B31(N,K1)M   (K1)A13(K1,N) + B32(N,K1)PHI   (K1)A23(K1,N)DAUX33
C                              -1                        -1             DAUX33
C              + B31(N,K2)M   (K2)A13(K2,N) + B32(N,K2)PHI   (K2)A23(K2,N)DAUX33
```

```
C                                                                      DAUX33
C                 - B35(N,N)                                           DAUX33
C                                                                      DAUX33
      IJ = IJ+1                                                        DAUX33
      IJK(NNS,MNS) = IJ                                                DAUX33
      IF (KQTYPE(N).EQ.2) GO TO 51                                     DAUX33
      IF (KQTYPE(N).EQ.4) GO TO 51                                     DAUX33
      DO 65 I=1,3                                                      DAUX33
      DO 65 J=1,3                                                      DAUX33
      SUM = -HHT(I,J,N)                                                DAUX33
      IF (I.EQ.J) SUM = 1.0+SUM                                        DAUX33
      DO 64 K=1,3                                                      DAUX33
   64 SUM = SUM + B31(I,K,2*N-1)*  RW(  K1)*A13(K,J,2*N-1)             DAUX33
     *            + B31(I,K,2*N  )*  RW(  K2)*A13(K,J,2*N  )           DAUX33
     *            + B32(I,K,2*N-1)*RPHI(K,K1)*A23(K,J,2*N-1)           DAUX33
     *            + B32(I,K,2*N  )*RPHI(K,K2)*A23(K,J,2*N  )           DAUX33
   65 C(I,J,IJ) = SUM                                                  DAUX33
      GO TO 59                                                         DAUX33
C                                                                      DAUX33
C     FOR KQTYPE = 2 OR 4, SET C33(N,N) = B*I                          DAUX33
C     WHERE B = SUM OF DIAGONAL ELEMENTS OF                            DAUX33
C             -1                  -1                                   DAUX33
C     (B31)(M)   (A13) + (B32)(PHI)   (A23)                            DAUX33
C                                                                      DAUX33
   51 SUM = 0.0                                                        DAUX33
      DO 55 I=1,3                                                      DAUX33
      DO 55 K=1,3                                                      DAUX33
   55 SUM = SUM + B31(I,K,2*N-1)*  RW(  K1)*A13(K,I,2*N-1)             DAUX33
     *            + B31(I,K,2*N  )*  RW(  K2)*A13(K,I,2*N  )           DAUX33
     *            + B32(I,K,2*N-1)*RPHI(K,K1)*A23(K,I,2*N-1)           DAUX33
     *            + B32(I,K,2*N  )*RPHI(K,K2)*A23(K,I,2*N  )           DAUX33
      DO 57 I=1,3                                                      DAUX33
      DO 56 J=1,3                                                      DAUX33
   56 C(I,J,IJ) = 0.0                                                  DAUX33
   57 C(I,I,IJ) = SUM                                                  DAUX33
   59 IF (N.EQ.NQ) GO TO 90                                            DAUX33
      N1 = N+1                                                         DAUX33
      DO 85 M=N1,NQ                                                    DAUX33
      IF (KQTYPE(M).LT.0) GO TO 85                                     DAUX33
      MNS = NQ2S - NQ + M                                              DAUX33
      IF  (ISING(K1).NE.0)  GO TO 75                                   DAUX33
      IF (K1.NE.KQ1(M))        GO TO 70                                DAUX33
      IF (IJK(MNS,NNS).NE.0) GO TO 67                                  DAUX33
C                                                                      DAUX33
C     FOR ANY M>N SUCH THAT K1(N) = K1(M)                              DAUX33
C                                                                      DAUX33
C                                         -1                           DAUX33
C     C33(N,M) = C(N,M) + B31(N,K1)   M   (K1)A13(K1,M)                DAUX33
C                                         -1                           DAUX33
C                         + B32(N,K1)PHI   (K1)A23(K1,M)               DAUX33
```

```
C                                                                       DAUX33
C                                       -1                              DAUX33
C     C33(M,N) = C(M,N) + B31(M,K1)  M   (K1)A13(K1,N)                  DAUX33
C                                       -1                              DAUX33
C                       + B32(M,K1)PHI   (K1)A23(K1,N)                  DAUX33
C                                                                       DAUX33
      IJ = IJ+1                                                         DAUX33
      IJK(MNS,NNS) = IJ                                                 DAUX33
      IJK(NNS,MNS) = IJ+1                                               DAUX33
      DO 66 J=1,3                                                       DAUX33
      DO 66 I=1,3                                                       DAUX33
      C(I,J,IJ  ) = 0.0                                                 DAUX33
   66 C(I,J,IJ+1) = 0.0                                                 DAUX33
      IJ = IJ+1                                                         DAUX33
   67 JJ = IJK(MNS,NNS)                                                 DAUX33
      DO 69 I=1,3                                                       DAUX33
      DO 69 J=1,3                                                       DAUX33
      SUM = C(I,J,JJ)                                                   DAUX33
      TUM = C(I,J,JJ+1)                                                 DAUX33
      DO 68 K=1,3                                                       DAUX33
      SUM = SUM + B31(I,K,2*N-1)*  RW(  K1)*A13(K,J,2*M-1)              DAUX33
     *          + B32(I,K,2*N-1)*RPHI(K,K1)*A23(K,J,2*M-1)              DAUX33
   68 TUM = TUM + B31(I,K,2*M-1)*  RW(  K1)*A13(K,J,2*N-1)              DAUX33
     *          + B32(I,K,2*M-1)*RPHI(K,K1)*A23(K,J,2*N-1)              DAUX33
      C(I,J,JJ  ) = SUM                                                 DAUX33
   69 C(I,J,JJ+1) = TUM                                                 DAUX33
   70 IF (K1.NE.KQ2(M))        GO TO 75                                 DAUX33
      IF (IJK(MNS,NNS).NE.0) GO TO 72                                   DAUX33
C                                                                       DAUX33
C     FOR ANY M>N SUCH THAT K1(N) = K2(M)                               DAUX33
C                                                                       DAUX33
C                                       -1                              DAUX33
C     C33(N,M) = C(N,M) + B31(N,K1)  M   (K1)A13(K2,M)                  DAUX33
C                                       -1                              DAUX33
C                       + B32(N,K1)PHI   (K1)A23(K2,M)                  DAUX33
C                                                                       DAUX33
C                                       -1                              DAUX33
C     C33(M,N) = C(M,N) + B31(M,K2)  M   (K1)A13(K1,N)                  DAUX33
C                                       -1                              DAUX33
C                       + B32(M,K2)PHI   (K1)A23(K1,N)                  DAUX33
C                                                                       DAUX33
      IJ = IJ+1                                                         DAUX33
      IJK(MNS,NNS) = IJ                                                 DAUX33
      IJK(NNS,MNS) = IJ+1                                               DAUX33
      DO 71 J=1,3                                                       DAUX33
      DO 71 I=1,3                                                       DAUX33
      C(I,J,IJ  ) = 0.0                                                 DAUX33
   71 C(I,J,IJ+1) = 0.0                                                 DAUX33
      IJ = IJ+1                                                         DAUX33
   72 JJ = IJK(MNS,NNS)                                                 DAUX33
```

```
      DO 74 I=1,3                                                       DAUX33
      DO 74 J=1,3                                                       DAUX33
      SUM = C(I,J,JJ)                                                   DAUX33
      TUM = C(I,J,JJ+1)                                                 DAUX33
      DO 73 K=1,3                                                       DAUX33
      SUM = SUM + B31(I,K,2*N-1)*  RW(  K1)*A13(K,J,2*M  )              DAUX33
     *          + B32(I,K,2*N-1)*RPHI(K,K1)*A23(K,J,2*M  )              DAUX33
   73 TUM = TUM + B31(I,K,2*N  )*  RW(  K1)*A13(K,J,2*M-1)              DAUX33
     *          + B32(I,K,2*N  )*RPHI(K,K1)*A23(K,J,2*M-1)              DAUX33
      C(I,J,JJ  ) = SUM                                                 DAUX33
   74 C(I,J,JJ+1) = TUM                                                 DAUX33
   75 IF  (ISING(K2).NE.0)  GO TO 85                                    DAUX33
      IF (K2.NE.KQ1(M))        GO TO 80                                 DAUX33
      IF (IJK(MNS,NNS).NE.0) GO TO 77                                   DAUX33
C                                                                       DAUX33
C     FOR ANY M>N SUCH THAT K2(N) = K1(M)                               DAUX33
C                                                                       DAUX33
C                                      -1                               DAUX33
C     C33(N,M) = C(N,M) + B31(N,K2)   M  (K2)A13(K1,M)                  DAUX33
C                                      -1                               DAUX33
C                       + B32(N,K2)PHI  (K2)A23(K1,M)                   DAUX33
C                                                                       DAUX33
C                                      -1                               DAUX33
C     C33(M,N) = C(M,N) + B31(M,K1)   M  (K2)A13(K2,N)                  DAUX33
C                                      -1                               DAUX33
C                       + B32(M,K1)PHI  (K2)A23(K2,N)                   DAUX33
C                                                                       DAUX33
      IJ = IJ+1                                                         DAUX33
      IJK(MNS,NNS) = IJ                                                 DAUX33
      IJK(NNS,MNS) = IJ+1                                               DAUX33
      DO 76 J=1,3                                                       DAUX33
      DO 76 I=1,3                                                       DAUX33
      C(I,J,IJ  ) = 0.0                                                 DAUX33
   76 C(I,J,IJ+1) = 0.0                                                 DAUX33
      IJ = IJ+1                                                         DAUX33
   77 JJ = IJK(MNS,NNS)                                                 DAUX33
      DO 79 I=1,3                                                       DAUX33
      DO 79 J=1,3                                                       DAUX33
      SUM = C(I,J,JJ)                                                   DAUX33
      TUM = C(I,J,JJ+1)                                                 DAUX33
      DO 78 K=1,3                                                       DAUX33
      SUM = SUM + B31(I,K,2*N  )*  RW(  K2)*A13(K,J,2*M-1)              DAUX33
     *          + B32(I,K,2*N  )*RPHI(K,K2)*A23(K,J,2*M-1)              DAUX33
   78 TUM = TUM + B31(I,K,2*M-1)*  RW(  K2)*A13(K,J,2*N  )              DAUX33
     *          + B32(I,K,2*M-1)*RPHI(K,K2)*A23(K,J,2*N  )              DAUX33
      C(I,J,JJ  ) = SUM                                                 DAUX33
   79 C(I,J,JJ+1) = TUM                                                 DAUX33
   80 IF (K2.NE.KQ2(M))      GO TO 85                                   DAUX33
      IF (IJK(MNS,NNS).NE.0) GO TO 82                                   DAUX33
C                                                                       DAUX33
```

```
C     FOR ANY M>N SUCH THAT K2(N) = K2(M)                             DAUX33
C                                                                     DAUX33
C                                           -1                        DAUX33
C     C33(N,M) = C(N,M) + B31(N,K2)  M   (K2)A13(K2,M)                DAUX33
C                                           -1                        DAUX33
C                         + B32(N,K2)PHI   (K2)A23(K2,M)              DAUX33
C                                                                     DAUX33
C                                           -1                        DAUX33
C     C33(M,N) = C(M,N) + B31(M,K2)  M   (K2)A13(K2,N)                DAUX33
C                                           -1                        DAUX33
C                         + B32(M,K2)PHI   (K2)A23(K2,N)              DAUX33
C                                                                     DAUX33
      IJ = IJ+1                                                       DAUX33
      IJK(MNS,NNS) = IJ                                               DAUX33
      IJK(NNS,MNS) = IJ+1                                             DAUX33
      DO 81 J=1,3                                                     DAUX33
      DO 81 I=1,3                                                     DAUX33
      C(I,J,IJ  ) = 0.0                                               DAUX33
   81 C(I,J,IJ+1) = 0.0                                               DAUX33
      IJ = IJ+1                                                       DAUX33
   82 JJ = IJK(MNS,NNS)                                               DAUX33
      DO 84 I=1,3                                                     DAUX33
      DO 84 J=1,3                                                     DAUX33
      SUM = C(I,J,JJ)                                                 DAUX33
      TUM = C(I,J,JJ+1)                                               DAUX33
      DO 83 K=1,3                                                     DAUX33
      SUM = SUM + B31(I,K,2*N  )*  RW(  K2)*A13(K,J,2*M  )            DAUX33
     *          + B32(I,K,2*N  )*RPHI(K,K2)*A23(K,J,2*M  )            DAUX33
   83 TUM = TUM + B31(I,K,2*M  )*  RW(  K2)*A13(K,J,2*N  )            DAUX33
     *          + B32(I,K,2*M  )*RPHI(K,K2)*A23(K,J,2*N  )            DAUX33
      C(I,J,JJ  ) = SUM                                               DAUX33
   84 C(I,J,JJ+1) = TUM                                               DAUX33
   85 CONTINUE                                                        DAUX33
   90 CONTINUE                                                        DAUX33
      CALL ELTIME(2,19)                                               DAUX33
      RETURN                                                          DAUX33
      END                                                             DAUX33
```

```
      SUBROUTINE DAUX44                                                 DAUX44
C                                                    REV IV    07/24/86SLIP
      IMPLICIT REAL*8(A-H,O-Z)                                          DAUX44
      COMMON/CONTRL/ TIME,NSEG,NJNT,NPL,NBLT,NBAG,NVEH,NGRND,           DAUX44
     *               NS,NQ,NSD,NFLX,NHRNSS,NWINDF,NJNTF,NPRT(36),NPG    PAGE
      COMMON/SGMNTS/ D(3,3,30),WMEG(3,30),WMEGD(3,30),U1(3,30),U2(3,30),DAUX44
     *               SEGLP(3,30),SEGLV(3,30),SEGLA(3,30),NSYM(30)       DAUX44
      COMMON/DESCRP/ PHI(3,30),W(30),RW(30),SR(4,60),HA(3,60),HB(3,60), SLIP
     *               RPHI(3,30),HT(3,3,60),SPRING(5,90),VISC(7,90),     DAUX44
     *               JNT(30),IPIN(30),ISING(30),IGLOB(30),JOINTF(30)    DAUX44
      COMMON/CMATRX/ V1(3,30),V2(3,30),V3(3,12),B12(3,3,60),A22(3,3,60),DAUX44
     *               F(3,30),TQ(3,30),WJ(30),A11(3,3,30)                SLIP
      COMMON/CSTRNT/ A13(3,3,24),A23(3,3,24),B31(3,3,24),B32(3,3,24),   DAUX44
     *               HHT(3,3,12),RK1(3,12),RK2(3,12),QQ(3,12),TQQ(3,12),DAUX44
     *               RQQ(3,12),HQQ(3,12),SQQ(12),CFQQ(12),              DAUX44
     *               KQ1(12),KQ2(12),KQTYPE(12)                         DAUX44
      COMMON/FLXBLE/ HF(4,12,8),B42(3,3,24),V4(3,8),NFLEX(3,8)          DAUX44
      LOGICAL*1 FREE                                                    SLIP
      COMMON/TEMPVS/ C(3,3,600),RHS(3,54),IJK(54,54),IJ,NQ2S            CHGIII
     *              ,IDUM(458),FREE(30)                                 SLIP
      IF (NFLX.EQ.0) GO TO 99                                           DAUX44
      CALL ELTIME(1,33)                                                 DAUX44
      DO 90 L=1,NFLX                                                    DAUX44
      N1 = NFLEX(1,L)                                                   DAUX44
      N2 = NFLEX(2,L)                                                   DAUX44
      N3 = NFLEX(3,L)                                                   DAUX44
      IJ = IJ+1                                                         DAUX44
      DO 10 I=1,3                                                       DAUX44
      DO 10 J=1,3                                                       DAUX44
      C(I,J,IJ) = 0.0                                                   DAUX44
      DO 10 K=1,3                                                       DAUX44
   10 C(I,J,IJ) = C(I,J,IJ) + B42(I,K,3*L-2)*RPHI(K,N1)*B42(J,K,3*L-2)  DAUX44
     *                      + B42(I,K,3*L-1)*RPHI(K,N2)*B42(J,K,3*L-1)  DAUX44
     *                      + B42(I,K,3*L  )*RPHI(K,N3)*B42(J,K,3*L  )  DAUX44
      NSL = 2*NS+L                                                      DAUX44
      IJK(NSL,NSL) = IJ                                                 DAUX44
      DO 20 I=1,3                                                       DAUX44
      RHS(I,NSL) = -V4(I,L)                                             DAUX44
      DO 20 J=1,3                                                       DAUX44
   20 RHS(I,NSL) = RHS(I,NSL)  + B42(I,J,3*L-2)*U2(I,N1)                DAUX44
     *                         + B42(I,J,3*L-1)*U2(I,N2)                DAUX44
     *                         + B42(I,J,3*L  )*U2(I,N3)                DAUX44
      IF (L.EQ.NFLX) GO TO 30                                           DAUX44
      LP1 = L+1                                                         DAUX44
      DO 29 M=LP1,NFLX                                                  DAUX44
      DO 28 II=1,3,2                                                    DAUX44
      IL = NFLEX(II,L)                                                  DAUX44
      IF  (ISING(IL).NE.0)  GO TO 28                                    DAUX44
      DO 27 JJ=1,3,2                                                    DAUX44
      IF (NFLEX(II,L).NE.NFLEX(JJ,M)) GO TO 27                          DAUX44
```

```
      NSM = 2*NS+M                                                      DAUX44
      JK = IJK(NSL,NSM)                                                 DAUX44
      KJ = IJK(NSM,NSL)                                                 DAUX44
      IF (JK.GT.0) GO TO 22                                             DAUX44
      IJK(NSL,NSM) = IJ+1                                               DAUX44
      IJK(NSM,NSL) = IJ+2                                               DAUX44
      JK = IJ+1                                                         DAUX44
      KJ = IJ+2                                                         DAUX44
      IJ = IJ+2                                                         DAUX44
      DO 21 I=1,3                                                       DAUX44
      DO 21 J=1,3                                                       DAUX44
   21 C(I,J,JK) = 0.0                                                   DAUX44
   22 LI = 3*L+II-3                                                     DAUX44
      MJ = 3*M+JJ-3                                                     DAUX44
      DO 24 I=1,3                                                       DAUX44
      DO 24 J=1,3                                                       DAUX44
      DO 23 K=1,3                                                       DAUX44
   23 C(I,J,JK) = C(I,J,JK) + B42(I,K,LI)*RPHI(K,IL)*B42(J,K,MJ)        DAUX44
   24 C(J,I,KJ) = C(I,J,JK)                                             DAUX44
   27 CONTINUE                                                          DAUX44
   28 CONTINUE                                                          DAUX44
   29 CONTINUE                                                          DAUX44
   30 IF (NQ.EQ.0) GO TO 40                                             DAUX44
      DO 39 M=1,NQ                                                      DAUX44
      IF (KQTYPE(M).LT.0) GO TO 39                                      DAUX44
      DO 38 II=1,3                                                      DAUX44
      LM = 0                                                            DAUX44
      IF (NFLEX(II,L).EQ.KQ1(M)) LM = 2*M-1                             DAUX44
      IF (NFLEX(II,L).EQ.KQ2(M)) LM = 2*M                               DAUX44
      IF (LM.EQ.0) GO TO 38                                             DAUX44
      IL = NFLEX(II,L)                                                  DAUX44
      IF  (ISING(IL).NE.0)  GO TO 38                                    DAUX44
      NSM = 2*NS+NFLX+M                                                 DAUX44
      JK = IJK(NSL,NSM)                                                 DAUX44
      KJ = IJK(NSM,NSL)                                                 DAUX44
      IF (JK.GT.0) GO TO 32                                             DAUX44
      IJK(NSL,NSM) = IJ+1                                               DAUX44
      IJK(NSM,NSL) = IJ+2                                               DAUX44
      JK = IJ+1                                                         DAUX44
      KJ = IJ+2                                                         DAUX44
      IJ = IJ+2                                                         DAUX44
      DO 31 I=1,3                                                       DAUX44
      DO 31 J=1,3                                                       DAUX44
      C(I,J,JK) = 0.0                                                   DAUX44
   31 C(I,J,KJ) = 0.0                                                   DAUX44
   32 LI = 3*L+II-3                                                     DAUX44
      DO 33 I=1,3                                                       DAUX44
      DO 33 J=1,3                                                       DAUX44
      DO 33 K=1,3                                                       DAUX44
      C(I,J,JK) = C(I,J,JK) + B42(I,K,LI)*RPHI(K,IL)*A23(K,J,LM)        DAUX44
```

```
   33 C(I,J,KJ) = C(I,J,KJ) + B32(I,K,LM)*RPHI(K,IL)*B42(J,K,LI)       DAUX44
   38 CONTINUE                                                            DAUX44
   39 CONTINUE                                                            DAUX44
   40 IF (NJNT.EQ.0) GO TO 90                                             DAUX44
      DO 59 M=1,NJNT                                                      DAUX44
      IF (JNT(M).EQ.0) GO TO 59                                           DAUX44
      DO 58 II=1,3                                                        DAUX44
      LM = 0                                                              DAUX44
      IF (NFLEX(II,L).EQ.IABS(JNT(M))) LM = 2*M-1                         DAUX44
      IF (NFLEX(II,L).EQ.M+1)          LM = 2*M                           DAUX44
      IF (LM.EQ.0) GO TO 58                                               DAUX44
      IL = NFLEX(II,L)                                                    DAUX44
      IF  (ISING(IL).NE.0)  GO TO 58                                      DAUX44
      NSM = 2*NS+NFLX+NQ+M                                                DAUX44
      JK = IJK(NSL,NSM)                                                   DAUX44
      KJ = IJK(NSM,NSL)                                                   DAUX44
      IF (JK.GT.0) GO TO 42                                               DAUX44
      IJK(NSL,NSM) = IJ+1                                                 DAUX44
      IJK(NSM,NSL) = IJ+2                                                 DAUX44
      JK = IJ+1                                                           DAUX44
      KJ = IJ+2                                                           DAUX44
      IJ = IJ+2                                                           DAUX44
      DO 41 I=1,3                                                         DAUX44
      DO 41 J=1,3                                                         DAUX44
   41 C(I,J,JK) = 0.0                                                     DAUX44
   42 LI = 3*L+II-3                                                       DAUX44
      DO 44 I=1,3                                                         DAUX44
      DO 44 J=1,3                                                         DAUX44
      DO 43 K=1,3                                                         DAUX44
   43 C(I,J,JK) = C(I,J,JK) + B42(I,K,LI)*RPHI(K,IL)*B12(J,K,LM)         DAUX44
   44 C(J,I,KJ) = C(I,J,JK)                                               DAUX44
      IF (FREE(M)) GO TO 58                                               SLIP
      NSM = 2*NS+NFLX+NQ+NJNT+M                                           DAUX44
      JK = IJK(NSL,NSM)                                                   DAUX44
      KJ = IJK(NSM,NSL)                                                   DAUX44
      IF (JK.GT.0) GO TO 52                                               DAUX44
      IJK(NSL,NSM) = IJ+1                                                 DAUX44
      IJK(NSM,NSL) = IJ+2                                                 DAUX44
      JK = IJ+1                                                           DAUX44
      KJ = IJ+2                                                           DAUX44
      IJ = IJ+2                                                           DAUX44
      DO 51 I=1,3                                                         DAUX44
      DO 51 J=1,3                                                         DAUX44
   51 C(I,J,JK) = 0.0                                                     DAUX44
   52 SET = 1.0                                                           DAUX44
      IF (IL.EQ.M+1) SET = -1.0                                           DAUX44
      DO 54 I=1,3                                                         DAUX44
      DO 54 J=1,3                                                         DAUX44
      DO 53 K=1,3                                                         DAUX44
   53 C(I,J,JK) = C(I,J,JK) + SET*B42(I,K,LI)*RPHI(K,IL)*A22(K,J,LM)     DAUX44
```

```
54 C(J,I,KJ) = C(I,J,JK)                                            DAUX44
58 CONTINUE                                                         DAUX44
59 CONTINUE                                                         DAUX44
90 CONTINUE                                                         DAUX44
   CALL ELTIME(2,33)                                                DAUX44
99 RETURN                                                           DAUX44
   END                                                              DAUX44
```

```
      SUBROUTINE DAUX55                                                 DAUX55
C                                                    REV IV    07/24/86SLIP
      IMPLICIT REAL*8(A-H,O-Z)                                          DAUX55
      COMMON/CONTRL/ TIME,NSEG,NJNT,NPL,NBLT,NBAG,NVEH,NGRND,           DAUX55
     *               NS,NQ,NSD,NFLX,NHRNSS,NWINDF,NJNTF,NPRT(36),NPG    PAGE
      COMMON/SGMNTS/ D(3,3,30),WMEG(3,30),WMEGD(3,30),U1(3,30),U2(3,30),DAUX55
     *               SEGLP(3,30),SEGLV(3,30),SEGLA(3,30),NSYM(30)       DAUX55
      COMMON/DESCRP/ PHI(3,30),W(30),RW(30),SR(4,60),HA(3,60),HB(3,60), SLIP
     *               RPHI(3,30),HT(3,3,60),SPRING(5,90),VISC(7,90),     DAUX55
     *               JNT(30),IPIN(30),ISING(30),IGLOB(30),JOINTF(30)    DAUX55
      COMMON/CMATRX/ V1(3,30),V2(3,30),V3(3,12),B12(3,3,60),A22(3,3,60),DAUX55
     *               F(3,30),TQ(3,30),WJ(30),A11(3,3,30)                SLIP
      COMMON/CSTRNT/ A13(3,3,24),A23(3,3,24),B31(3,3,24),B32(3,3,24),   DAUX55
     *               HHT(3,3,12),RK1(3,12),RK2(3,12),QQ(3,12),TQQ(3,12),DAUX55
     *               RQQ(3,12),HQQ(3,12),SQQ(12),CFQQ(12),              DAUX55
     *               KQ1(12),KQ2(12),KQTYPE(12)                         DAUX55
      COMMON/FLXBLE/ HF(4,12,8),B42(3,3,24),V4(3,8),NFLEX(3,8)          DAUX55
      COMMON/CNSNTS/ PI,RADIAN,G,THIRD,EPS(24),                         DAUX55
     *               UNITL,UNITM,UNITT,GRAVTY(3),TWOPI                  TWOPI
      LOGICAL*1 FREE                                                    SLIP
      COMMON/TEMPVS/ C(3,3,600),RHS(3,54),IJK(54,54),IJ,NQ2S            CHGIII
     *               ,IDUM(458),FREE(30)                                SLIP
      CALL ELTIME(1,30)                                                 DAUX55
      IS = 0                                                            DAUX55
      DO  99  I=1,NGRND                                                 DAUX55
      IF  (ISING(I).LE.0)  GO TO 99                                     DAUX55
      IS = IS+1                                                         DAUX55
      IJ = IJ+1                                                         DAUX55
      IJK(IS  ,IS  ) = IJ                                               DAUX55
      IJK(IS+1,IS+1) = IJ+1                                             DAUX55
      DO 11 J=1,3                                                       DAUX55
      RHS(J,IS  ) = U1(J,I) + W(I)*GRAVTY(J)/G                          DAUX55
      RHS(J,IS+1) = U2(J,I)                                             DAUX55
      U1(J,I) = 0.0                                                     DAUX55
      U2(J,I) = 0.0                                                     DAUX55
      DO 10 K=1,3                                                       DAUX55
      C(J,K,IJ  ) = 0.0                                                 DAUX55
   10 C(J,K,IJ+1) = 0.0                                                 DAUX55
      C(J,J,IJ  ) = W(I)/G                                              DAUX55
   11 C(J,J,IJ+1) = PHI(J,I)                                            DAUX55
      IJ = IJ+1                                                         DAUX55
      IF (NFLX.EQ.0) GO TO 19                                           DAUX55
      DO 15 N=1,NFLX                                                    DAUX55
      LN = 0                                                            DAUX55
      IF (NFLEX(1,N).EQ.I) LN = 3*N-2                                   DAUX55
      IF (NFLEX(2,N).EQ.I) LN = 3*N-1                                   DAUX55
      IF (NFLEX(3,N).EQ.I) LN = 3*N                                     DAUX55
      IF (LN.EQ.0) GO TO 15                                             DAUX55
      DO 14 J=1,3                                                       DAUX55
      DO 14 K=1,3                                                       DAUX55
```

```
      C(J,K,IJ+1) =  B42(K,J,LN)                                        DAUX55
   14 C(J,K,IJ+2) =  B42(J,K,LN)                                        SLIP
      NNS = 2*NS+N                                                      DAUX55
      IJK(IS+1,NNS) = IJ+1                                              DAUX55
      IJK(NNS,IS+1) = IJ+2                                              DAUX55
      IJ = IJ+2                                                         DAUX55
   15 CONTINUE                                                          DAUX55
   19 IF (NQ.EQ.0) GO TO 30                                             DAUX55
      DO 25 N=1,NQ                                                      DAUX55
      IF (KQTYPE(N).LT.0) GO TO 25                                      DAUX55
      LN = 0                                                            DAUX55
      IF (I.EQ.KQ1(N)) LN = 2*N-1                                       DAUX55
      IF (I.EQ.KQ2(N)) LN = 2*N                                         DAUX55
      IF (LN.EQ.0) GO TO 25                                             DAUX55
      DO 20 J=1,3                                                       DAUX55
      DO 20 K=1,3                                                       DAUX55
      C(J,K,IJ+1) =  A13(J,K,LN)                                        DAUX55
      C(J,K,IJ+2) =  A23(J,K,LN)                                        DAUX55
      C(J,K,IJ+3) =  B31(J,K,LN)                                        SLIP
   20 C(J,K,IJ+4) =  B32(J,K,LN)                                        SLIP
      NNS = 2*NS+NFLX+N                                                 DAUX55
      IJK(IS  ,NNS) = IJ+1                                              DAUX55
      IJK(IS+1,NNS) = IJ+2                                              DAUX55
      IJK(NNS,IS  ) = IJ+3                                              DAUX55
      IJK(NNS,IS+1) = IJ+4                                              DAUX55
      IJ = IJ+4                                                         DAUX55
   25 CONTINUE                                                          DAUX55
   30 IF (NJNT.EQ.0) GO TO 98                                           DAUX55
      DO 65 N=1,NJNT                                                    DAUX55
      IF (JNT(N).EQ.0) GO TO 65                                         DAUX55
      LN = 0                                                            DAUX55
      IF (I.EQ.IABS(JNT(N))) LN = 2*N-1                                 DAUX55
      IF (I.EQ.N+1)          LN = 2*N                                   DAUX55
      IF (LN.EQ.0) GO TO 65                                             DAUX55
      SET = 1.0                                                         DAUX55
      IF (I.EQ.N+1) SET = -1.0                                          DAUX55
      DO 40 J=1,3                                                       DAUX55
      DO 40 K=1,3                                                       SLIP
      C(J,K,IJ+1) = SET*A11(J,K,N)                                      SLIP
      C(J,K,IJ+3) = SET*A11(K,J,N)                                      SLIP
      C(J,K,IJ+2) =  B12(K,J,LN)                                        DAUX55
   40 C(J,J,IJ+4) = B12(J,K,LN)                                         SLIP
      NNS = NQ2S + N                                                    DAUX55
      IJK(IS  ,NNS) = IJ+1                                              DAUX55
      IJK(IS+1,NNS) = IJ+2                                              DAUX55
      IJK(NNS,IS  ) = IJ+3                                              DAUX55
      IJK(NNS,IS+1) = IJ+4                                              DAUX55
      IJ = IJ+4                                                         DAUX55
      IF (FREE(N)) GO TO 65                                             SLIP
      DO 60 J=1,3                                                       DAUX55
```

```
      DO 60 K=1,3                                              DAUX55
      C(J,K,IJ+1) =  SET*A22(J,K,LN)                           DAUX55
   60 C(J,K,IJ+2) =  SET*A22(K,J,LN)                           SLIP
      NNS = NQ2S + NJNT + N                                    DAUX55
      IJK(IS+1,NNS) = IJ+1                                     DAUX55
      IJK(NNS,IS+1) = IJ+2                                     DAUX55
      IJ = IJ+2                                                DAUX55
   65 CONTINUE                                                 DAUX55
   98 IS = IS+1                                                DAUX55
   99 CONTINUE                                                 DAUX55
      CALL ELTIME(2,30)                                        DAUX55
      RETURN                                                   DAUX55
      END                                                      DAUX55
```

```
      SUBROUTINE DHHPIN(DD,BN,L,M,N)                                    DHHPIN
C                                                     REV IV    07/24/86SLIP
C     SETS DD = D(L) IF JOINT M IS NOT PINNED                           DHHPIN
C       OR DD = (I-HH.)(D(L)) IF PINNED                                 DHHPIN
C                                                                       DHHPIN
      IMPLICIT REAL*8 (A-H,O-Z)                                         DHHPIN
      COMMON/SGMNTS/ D(3,3,30),WMEG(3,30),WMEGD(3,30),U1(3,30),U2(3,30),DHHPIN
     *               SEGLP(3,30),SEGLV(3,30),SEGLA(3,30),NSYM(30)       DHHPIN
      COMMON/DESCRP/ PHI(3,30),W(30),RW(30),SR(4,60),HA(3,60),HB(3,60), SLIP
     *               RPHI(3,30),HT(3,3,60),SPRING(5,90),VISC(7,90),     DHHPIN
     *               JNT(30),IPIN(30),ISING(30),IGLOB(30),JOINTF(30)    DHHPIN
      COMMON/CEULER/ IEULER(30),HIR(3,3,90),ANG(3,30),ANGD(3,30),       JDRIFT
     *               FE(3,30),TQE(3,30),CONST(5,30)                     JDRIFT
      DIMENSION DD(3,3),BN(3)                                           DHHPIN
      DO 10 J=1,3                                                       DHHPIN
      BN(J) = 0.0                                                       DHHPIN
      DO 10 I=1,3                                                       DHHPIN
   10 DD(I,J) = D(I,J,L)                                                DHHPIN
      LGO = IPIN(M)+8                                                   SLIP
      TSIGN = -1.0                                                      DHHPIN
      GO TO (90,90,90,20,90,90,90,90,30,90,90,90,90,30,30),LGO          SLIP
   20 IF (IEULER(M).GE.7) GO TO 90                                      DHHPIN
      IF (IEULER(M).GE.4) GO TO 30                                      DHHPIN
      TSIGN = 1.0                                                       DHHPIN
      DO 21 J=1,3                                                       DHHPIN
      DO 21 I=1,3                                                       DHHPIN
   21 DD(I,J) = 0.0                                                     DHHPIN
   30 DO 31 J=1,3                                                       DHHPIN
      BN(J) = HB(1,N)*D(1,J,L) + HB(2,N)*D(2,J,L) + HB(3,N)*D(3,J,L)    DHHPIN
      DO 31 I=1,3                                                       DHHPIN
   31 DD(I,J) = DD(I,J) + TSIGN*BN(J)*HB(I,N)                           DHHPIN
   90 RETURN                                                            DHHPIN
      END                                                               DHHPIN
```

```
      SUBROUTINE DINT                                                   DINT
C                                                 REV IV    07/23/86TWOPI
      IMPLICIT REAL*8 (A-H,O-Z)                                         DINT
      COMMON/CONTRL/ TIME,NSEG,NJNT,NPL,NBLT,NBAG,NVEH,NGRND,           DINT
     *               NS,NQ,NSD,NFLX,NHRNSS,NWINDF,NJNTF,NPRT(36),NPG    PAGE
      COMMON/INTEST/ SGTEST(3,4,30),XTEST(360  ),SEGT(120),REGT(120)    DINT
C     NOTE: XTEST SINGLY DIMENSIONED HERE.                              DINT
      REAL    SEGT                                                      DINT
      COMMON/CNSNTS/ PI,RADIAN,G,THIRD,EPS(24),                         DINT
     *               UNITL,UNITM,UNITT,GRAVTY(3),TWOPI                  TWOPI
      COMMON/CDINT/ UU(4),GH(3,4),                                      DINT
     *              E(3,240), F(5,240),GG(5,240),Y(5,240),U(5,240),     DINT
     *              H,HPRINT,HS,TPRINT,TSTART,ICNT,IDBL,IFLAG           DINT
      COMMON/COMAIN/ VAR(240),DER(240),DT,H0,HMAX,HMIN,RSTIME,          DINT
     *               ISTEP,NSTEPS,NDINT,NEQ,IRSIN,IRSOUT                DINT
      LOGICAL LNRT                                                      TGMOD1
      CALL ELTIME(1,3)                                                  DINT
      IF  (ISTEP.NE.0)  GO TO 11                                        DINT
C                                                                       DINT
C     IN=0: INITIAL CALL TO INTEGRATOR - INITIALIZE AND RESET PARAMETERSDINT
C     NOTE: FOR EARLIER VERSIONS OF CVS, THE VARIABLE 'IN'(ISTEP IN THE DINT
C           CALLING PROGRAM) RAN FRON 1 TO NSTEPS+1, NOW IT RUNS FROM   DINT
C           0 TO NSTEPS.                                                DINT
C                                                                       DINT
      TPRINT = TIME                                                     DINT
      IDBL = 2                                                          DINT
      K = 0                                                             DINT
      GO TO 13                                                          DINT
C                                                                       DINT
C     IN*0: ADVANCE TPRINT - TIME TO RETURN TO CALLING PROGRAM.         DINT
C                                                                       DINT
   11 TPRINT = TPRINT + DT                                              DINT
      H = HPRINT                                                        DINT
C                                                                       DINT
C     ENTRY TO ADVANCE INTEGRATOR                                       DINT
C                                                                       DINT
   12 K = 1                                                             DINT
      CALL UPDATE(K)                                                    DINT
C                                                                       DINT
C      NEGATIVE K FROM UPDATE IS INDICATOR TO RESET INTEGRAROR.         DINT
C                                                                       DINT
      IF  (K.EQ.1)  GO TO 15                                            DINT
C                                                                       DINT
C      RESET OR INITIALIZE INTEGRATOR.                                  DINT
C                                                                       DINT
   13 H = H0                                                            DINT
      HPRINT = H0                                                       DINT
      HS = 0.0                                                          DINT
      ICNT = -2                                                         DINT
      IF  (ISTEP.EQ.0  .OR. NPRT(26).EQ.2)  CALL OUTPUT(0)              DINT
```

```
      CALL PDAUX (VAR,DER,NEQ,K)                                      DINT
      IF  (ISTEP.NE.0 .AND. NPRT(26).EQ.2)  CALL OUTPUT(1)            DINT
      DO  14  I=1,NEQ                                                 DINT
      F(1,I) = VAR(I)                                                 DINT
      F(2,I) = DER(I)                                                 DINT
      DO  14  J=3,5                                                   DINT
      F(J,I) = 0.0                                                    DINT
      U(J,I) = 0.0                                                    DINT
   14 Y(J,I) = 0.0                                                    DINT
      IF  (ISTEP.EQ.0)  GO TO 65                                      DINT
      K = 1                                                           DINT
C                                                                     DINT
C     ADJUST H (CURRENT TIME STEP) IF IT WILL ADVANCE T BEYOND TPRINT. DINT
C                                                                     DINT
   15 IF  (H+EPS(8).GE.TPRINT-TIME)  H = TPRINT-TIME                  DINT
C                                                                     DINT
C     BACKUP ENTRY POINT IF H HAS BEEN HALVED.                        DINT
C                                                                     DINT
   16 D1 = 0.5*H                                                      DINT
      CALL TRIGFS                                                     DINT
      TSTART = TIME                                                   DINT
      DO  20  I=1,NEQ                                                 DINT
      U(3,I) = Y(5,I)                                                 DINT
      U(4,I) = U(5,I)                                                 DINT
      DO  20  J=1,5                                                   DINT
   20 GG(J,I) = F(J,I)                                                DINT
      CALL CMPUTE (K,1,D1)                                            DINT
      IF (K.LT.0)  GO TO 50                                           DINT
      CALL ADJUST   (1,D1)                                            DINT
      K = 2                                                           DINT
      CALL CMPUTE (K,0,D1)                                            DINT
      IF (K.LT.0)  GO TO 50                                           DINT
      CALL ADJUST   (2,D1)                                            DINT
      NQUAT = K                                                       DINT
      K = 3                                                           DINT
      CALL CMPUTE (K,1, H)                                            DINT
      IF (K.LT.0)  GO TO 50                                           DINT
      CALL ADJUST   (3,D1)                                            DINT
      DO  49  L=1,NDINT                                               DINT
      M = 1                                                           DINT
      IF  (L.EQ.1)  M = 0                                             DINT
      IF  (NPRT(26).NE.2)  CALL OUTPUT(0)                             DINT
      CALL CMPUTE (K,M, H)                                            DINT
      IF (K.LT.0)  GO TO 50                                           DINT
      FAIL = 1.0                                                      DINT
      JJ = 0                                                          DINT
      DO  47  II=1,NEQ,3                                              DINT
      JJ = JJ+1                                                       DINT
      IF  (XTEST(II).LE.0.0)  GO TO 47                                DINT
      TT = DER(II)**2 + DER(II+1)**2 + DER(II+2)**2                   DINT
```

```
      TX = VAR(II)**2 + VAR(II+1)**2 + VAR(II+2)**2                     DINT
      TE = 0.0                                                          DINT
      TY = 0.0                                                          DINT
      I2 = II+2                                                         DINT
      DO  45  I=II,I2                                                   DINT
      Z = GG(5,I)*(VAR(I)-GG(1,I)) + GG(2,I) + H*(GG(3,I)+H*GG(4,I))    DINT
      TE = TE + (DER(I)-Z)**2                                           DINT
      TYD = TT + TX*GG(5,I)**2                                          DINT
      IF  (TYD.EQ.0.0)  TYD = 1.0                                       DINT
   45 TY = TY + (DER(I)-Z)**2/TYD                                       DINT
      TM = 1000.0*TIME                                                  DINT
      IF  (NPRT(25).NE.0)  WRITE (6,46) TM,SEGT(JJ),REGT(JJ),TT,TE,TY,  DINT
     *                                 (XTEST(I),I=II,I2)               DINT
   46 FORMAT ('0 DINT CONV. TEST',F10.3,2X,A4,2X,A8,6G12.4)             DINT
      IF  (TT.LT.XTEST(II))  GO TO 47                                   DINT
      IF  (XTEST(II+1).GT.0.0 .AND. TE.LT.XTEST(II+1))  GO TO 47        DINT
      IF  (TY.GT.XTEST(II+2))  GO TO 48                                 DINT
   47 CONTINUE                                                          DINT
      FAIL = 0.0                                                        DINT
   48 CALL ADJUST   (4,D1)                                              DINT
      IF  (FAIL.EQ.0.0)  GO TO 60                                       DINT
      IF  (L.EQ.NDINT)   GO TO 49                                       DINT
      CALL CMPUTE (K,1,D1)                                              DINT
      IF  (K.LT.0)       GO TO 50                                       DINT
      CALL ADJUST   (5,D1)                                              DINT
   49 CONTINUE                                                          DINT
      IF  (NPRT(25).EQ.0)  WRITE (6,46) TM,SEGT(JJ),REGT(JJ),TT,TE,TY,  DINT
     *                                 (XTEST(I),I=II,I2)               DINT
   50 WRITE  (6,51)  TIME,H                                             DINT
   51 FORMAT('0 TEST FAILED AT TIME = ',F10.6,' FOR H = ',F10.6)        DINT
      ICNT = 0                                                          DINT
      IDBL = IDBL+2                                                     DINT
      IF  (IDBL.GT.6)  IDBL = 6                                         DINT
      IF  (K.GE.0)  GO TO 58                                            DINT
      IF  (H.GT.HMIN+EPS(8))  GO TO 59                                  DINT
      WRITE  (6,52)                                                     DINT
   52 FORMAT('0 PROGRAM TERMINATED. PDAUX NEG SQRT. H < HMIN+EPS8.'/    DINT
     *       '  RERUN PROGRAM WITH SMALLER HMIN ON INPUT CARD A.4')     DINT
      STOP 31                                                           DINT
   58 IF  (H.LE.HMIN+EPS(8))  GO TO 61                                  DINT
      IF  (NPRT(26).EQ.2)  CALL OUTPUT(1)                               DINT
   59 TIME = TSTART                                                     DINT
      H = 0.5*H                                                         DINT
      HPRINT = 0.5*HPRINT                                               DINT
      K = 2                                                             DINT
      GO TO 16                                                          DINT
   60 IF  (H.GT.0.74*HPRINT)  ICNT = ICNT+1                             DINT
   61 K = 4                                                             DINT
      M = 0                                                             DINT
      IF  (H.GT.HMIN .AND. IDBL.GT.2)  IDBL = IDBL-1                    DINT
```

```
      GG4 = 2.0*H                                                     DINT
      GG5 = DEXP(-1600.0*H)                                           DINT
      DO  63  I=1,NEQ                                                 DINT
      F(3,I) = GG(3,I) + GG4*GG(4,I)                                  DINT
      F(4,I) = GG(4,I)                                                DINT
      F(5,I) = GG(5,I)                                                DINT
      Y(3,I) = Y(1,I)                                                 DINT
      Y(4,I) = Y(2,I)                                                 DINT
      Y(5,I) = GG5*U(3,I)                                             DINT
   63 U(5,I) = GG5*U(4,I)                                             DINT
      CALL QSET(F,Y,VAR,DER,NQUAT)                                    DINT
      CALL PDAUX (VAR,DER,M,K)                                        DINT
      DO  64  I=1,NEQ                                                 DINT
      F(1,I) = VAR(I)                                                 DINT
   64 F(2,I) = DER(I)                                                 DINT
      HS = H                                                          DINT
      IF  (ICNT.LT.IDBL)  GO TO 65                                    DINT
      ICNT = 0                                                        DINT
      H = DMIN1(2.0*H,HMAX)                                           DINT
      HPRINT = DMIN1(2.0*HPRINT,HMAX)                                 DINT
   65 CALL UPDATE(2)                                                  DINT
      XPRINT = TPRINT - TIME                                          TGMOD1
      IF(XPRINT.GE.EPS(8).AND.NPRT(26).NE.3.AND.NPRT(26).GE.0)        TGMOD1
     * CALL OUTPUT(1)                                                 TGMOD1
      IF(XPRINT.GE.EPS(8)) GO TO 12                                   TGMOD1
      LNRT = .FALSE.                                                  TGMOD1
      IF(NPRT(26).GE.0) LNRT = .TRUE.                                 TGMOD1
      IF(NPRT(26).LT.0) INRT = IABS(NPRT(26))                         TGMOD1
      IF(NPRT(26).LT.0) LNRT = (MOD(ISTEP,INRT).EQ.0)                 TGMOD1
      IF(LNRT) CALL OUTPUT(1)                                         TGMOD1
      CALL ELTIME(2,3)                                                DINT
      RETURN                                                          DINT
      END                                                             DINT
```

```
      SUBROUTINE DOTT31 (A,B,C)                                         DOTT31
C                                                 REV 17    12/20/76DOTT31
C     PERFORMS MATRIX MULTIPLICATION C = AB'                            DOTT31
C     WHERE C IS A 3X3 MATRIX, AND A AND B ARE VECTORS OF LENGTH 3.     DOTT31
C                                                                       DOTT31
      IMPLICIT REAL*8 (A-H,O-Z)                                         DOTT31
      DIMENSION  A(3) , B(3) , C(3,3)                                   DOTT31
      DO  10  I=1,3                                                     DOTT31
      DO  10  J=1,3                                                     DOTT31
   10 C(I,J) = A(I)*B(J)                                                DOTT31
      RETURN                                                            DOTT31
      END                                                               DOTT31
```

```
      SUBROUTINE DOTT33 (A,B,C)                                         DOTT33
C                                                 REV 17    01/03/77DOTT33
C     PERFORMS MATRIX MULTIPLICATION C = AB'                            DOTT33
C     WHERE A, B AND C ARE ALL 3X3 MATRICEES.                           DOTT33
C                                                                       DOTT33
      IMPLICIT REAL*8 (A-H,O-Z)                                         DOTT33
      DIMENSION  A(3,3) , B(3,3) , C(3,3)                               DOTT33
      DO  10  I=1,3                                                     DOTT33
      DO  10  J=1,3                                                     DOTT33
   10 C(I,J) = A(I,1)*B(J,1) + A(I,2)*B(J,2) + A(I,3)*B(J,3)            DOTT33
      RETURN                                                            DOTT33
      END                                                               DOTT33
```

```
      SUBROUTINE DOT31 (A,B,C)                                          DOT31
C                                                   REV 17    01/03/77DOT31
C     PERFORMS MATRIX MULTIPLICATION C = A'B                            DOT31
C     WHERE A IS A 3X3 MATRIX, AND B AND C ARE VECTORS OF LENGTH 3.     DOT31
C                                                                       DOT31
      IMPLICIT REAL*8 (A-H,O-Z)                                         DOT31
      DIMENSION  A(3,3) , B(3) , C(3)                                   DOT31
      C(1) = A(1,1)*B(1) + A(2,1)*B(2) + A(3,1)*B(3)                    DOT31
      C(2) = A(1,2)*B(1) + A(2,2)*B(2) + A(3,2)*B(3)                    DOT31
      C(3) = A(1,3)*B(1) + A(2,3)*B(2) + A(3,3)*B(3)                    DOT31
      RETURN                                                            DOT31
      END                                                               DOT31
```

```
      SUBROUTINE DOT33 (A,B,C)                                          DOT33
C                                                   REV 17    01/03/77DOT33
C     PERFORMS MATRIX MULTIPLICATION C = A'B                            DOT33
C     WHERE A, B AND C ARE ALL 3X3 MATRICEES.                           DOT33
C                                                                       DOT33
      IMPLICIT REAL*8 (A-H,O-Z)                                         DOT33
      DIMENSION  A(3,3) , B(3,3) , C(3,3)                               DOT33
      DO  10  I=1,3                                                     DOT33
      DO  10  J=1,3                                                     DOT33
   10 C(I,J) = A(1,I)*B(1,J) + A(2,I)*B(2,J) + A(3,I)*B(3,J)            DOT33
      RETURN                                                            DOT33
      END                                                               DOT33
```

```
      SUBROUTINE DRCIJK (D,ANG,ID,HT,J)                                 DRCIJK
C                                                   REV 18    02/24/78DRCIJK
      IMPLICIT REAL*8 (A-H,O-Z)                                         DRCIJK
      DIMENSION  D(9,22),HT(9,42),ANG(3,22),ID(4,22),T1(9),T2(9)        DRCIJK
      M = ID(4,J)                                                       DRCIJK
      IF  (M.NE.0)  GO TO 10                                            DRCIJK
      CALL DRCYPR (D(1,J),ANG(1,J),ID(1,J))                             DRCIJK
      GO TO 99                                                          DRCIJK
   10 CALL DRCYPR (T1,ANG(1,J),ID(1,J))                                 DRCIJK
      IF  (M.LT.0)  GO TO 20                                            DRCIJK
      CALL MAT33 (T1,D(1,M),D(1,J))                                     DRCIJK
      GO TO 99                                                          DRCIJK
   20 M = -M                                                            DRCIJK
      CALL DOT33 (HT(1,2*J-3),D(1,M),D(1,J))                            DRCIJK
      CALL MAT33 (T1,D(1,J),T2)                                         DRCIJK
      CALL MAT33 (HT(1,2*J-2),T2,D(1,J))                                DRCIJK
   99 RETURN                                                            DRCIJK
      END                                                               DRCIJK
```

```
      SUBROUTINE DRCQUA(DC,Q)                                          DRCQUA
C                                                     REV III.5 07/31/85JTF785
C     COMPUTES DIRECTION COSINE MATRIX FROM QUATERNIONS                DRCQUA
      IMPLICIT REAL*8(A-H,O-Z)                                         DRCQUA
      DIMENSION DC(3,3),Q(4)                                           DRCQUA
      C = Q(1)**2 - Q(2)**2 - Q(3)**2 - Q(4)**2                        JTF785
      DO 12 I = 1,3                                                    DRCQUA
      DO 10 J = 1,3                                                    DRCQUA
   10 DC(I,J) = 2.0*Q(I+1)*Q(J+1)                                      DRCQUA
   12 DC(I,I) = DC(I,I) + C                                            DRCQUA
      E = Q(1) + Q(1)                                                  DRCQUA
      DO 14 I = 1,3                                                    DRCQUA
      J = 1 + MOD(I,3)                                                 DRCQUA
      K = 1 + MOD(I+1,3)                                               DRCQUA
      D = E*Q(I+1)                                                     DRCQUA
      DC(K,J) = DC(K,J) - D                                            DRCQUA
   14 DC(J,K) = DC(J,K) + D                                            DRCQUA
      DO 18 I =1,3                                                     DRCQUA
      DO 18 J = 1,3                                                    DRCQUA
   18 IF(DABS(DC(I,J)).GT.1.0D0)DC(I,J) = DSIGN(1.0D0,DC(I,J))         DRCQUA
      RETURN                                                           DRCQUA
      END                                                              DRCQUA
```

```
      SUBROUTINE DRCYPR (D,A,ID)                                        DRCYPR
C                                               REV IV    07/23/86TWOPI
C     SETS UP 3X3 DIRECTION COSINE MATRIX FOR GIVEN YAW,PITCH AND ROLL. DRCYPR
C                                                                       DRCYPR
C       ARGUMENTS:                                                      DRCYPR
C         D: 3X3 DIRECTION COSINE MATRIX TO BE COMPUTED.                DRCYPR
C         A: ARRAY OF LENGTH 3 CONTAINING ROTATATION ANGLES (DEGREES).  DRCYPR
C        I1: AXIS OF ROTATION FOR 1ST ANGLE (1,2,3 = X,Y,Z)             DRCYPR
C        I2: AXIS OF ROTATION FOR 2ND ANGLE (1,2,3 = X,Y,Z)             DRCYPR
C        I3: AXIS OF ROTATION FOR 3RD ANGLE (1,2,3 = X,Y,Z)             DRCYPR
C                                                                       DRCYPR
      IMPLICIT REAL*8 (A-H,O-Z)                                         DRCYPR
      COMMON/CNSNTS/ PI,RADIAN,G,THIRD,EPS(24),                         DRCYPR
     *               UNITL,UNITM,UNITT,GRAVTY(3),TWOPI                  TWOPI
      DIMENSION D(3,3),A(3),ID(3),T(3,3),B(3),S(3)                      DRCYPR
      IDSUM = ID(1) + ID(2) + ID(3)                                     DRCYPR
      DO  12  I=1,3                                                     DRCYPR
      B(I) = A(I)*RADIAN                                                DRCYPR
      DO  11  J=1,3                                                     DRCYPR
   11 D(I,J) = 0.0                                                      DRCYPR
   12 D(I,I) = 1.0                                                      DRCYPR
      DO  30  N=1,3                                                     DRCYPR
      IDN = IABS(ID(N))                                                 DRCYPR
      M = 4 - IDN                                                       DRCYPR
      IF  (ID(N).LT.0)  M = IDSUM - ID(N) - 2                           DRCYPR
      IF  (B(M).EQ.0.0)  GO TO 30                                       DRCYPR
      CALL ROT (T,IDN,B(M))                                             DRCYPR
      DO  23  J=1,3                                                     DRCYPR
      DO  21  K=1,3                                                     DRCYPR
      S(K) = D(K,J)                                                     DRCYPR
   21 D(K,J) = 0.0                                                      DRCYPR
      DO  22  I=1,3                                                     DRCYPR
      DO  22  K=1,3                                                     DRCYPR
   22 D(I,J) = D(I,J) + T(I,K)*S(K)                                     DRCYPR
   23 CONTINUE                                                          DRCYPR
   30 CONTINUE                                                          DRCYPR
      RETURN                                                            DRCYPR
      END                                                               DRCYPR
```

```
      SUBROUTINE DRIFT                                                  DRIFT
C                                                  REV IV    07/24/86SLIP
C     CORRECTS FOR DRIFT IN CONSTRAINED JOINTS                          DRIFT
C                                                                       DRIFT
C                                                                       DRIFT
      IMPLICIT REAL*8(A-H,O-Z)                                          DRIFT
      COMMON/CONTRL/ TIME,NSEG,NJNT,NPL,NBLT,NBAG,NVEH,NGRND,           DRIFT
     *               NS,NQ,NSD,NFLX,NHRNSS,NWINDF,NJNTF,NPRT(36),NPG    DRIFT
      COMMON/SGMNTS/ D(3,3,30),WMEG(3,30),WMEGD(3,30),U1(3,30),U2(3,30),DRIFT
     *               SEGLP(3,30),SEGLV(3,30),SEGLA(3,30),NSYM(30)       DRIFT
      COMMON/DESCRP/ PHI(3,30),W(30),RW(30),SR(4,60),HA(3,60),HB(3,60), SLIP
     *               RPHI(3,30),HT(3,3,60),SPRING(5,90),VISC(7,90),     DRIFT
     *               JNT(30),IPIN(30),ISING(30),IGLOB(30),JOINTF(30)    DRIFT
      COMMON/CEULER/ IEULER(30),HIR(3,3,90),ANG(3,30),ANGD(3,30),       DRIFT
     *               FE(3,30),TQE(3,30),CONST(5,30)                     DRIFT
      COMMON/CNSNTS/ PI,RADIAN,G,THIRD,EPS(24),                         DRIFT
     *               UNITL,UNITM,UNITT,GRAVTY(3),TWOPI                  TWOPI
      COMMON/TEMPVS/ T1(3),T2(3),T3(3),T4(3),TP(3,3),H1(3),H2(3)        DRIFT
      IF (NJNT.EQ.0) GO TO 51                                           DRIFT
      DO 50 J=1,NJNT                                                    DRIFT
      K = IABS(JNT(J))                                                  DRIFT
      IF (K.EQ.0) GO TO 50                                              DRIFT
      IF (ISING(J+1).LT.0) GO TO 50                                     DRIFT
C                                                                       DRIFT
      M = 0                                                             DRIFT
      IF (IPIN(J).EQ.1)   M = 4                                         DRIFT
      IF (IPIN(J).EQ.6)  M = 4                                          SLIP
      IF (IPIN(J).EQ.7)  M = 4                                          SLIP
      IF (IABS(IPIN(J)).NE.4)  GO TO 15                                 DRIFT
      IF (IEULER(J).EQ.1)  M = 2                                        DRIFT
      IF (IEULER(J).EQ.2)  M = 3                                        DRIFT
      IF (IEULER(J).EQ.3)  M = 1                                        DRIFT
      IF (IEULER(J).EQ.4)  M = 4                                        DRIFT
      IF (IEULER(J).EQ.5)  M = 4                                        DRIFT
      IF (IEULER(J).EQ.6)  M = 4                                        DRIFT
   15 IF (M.EQ.0) GO TO 50                                              DRIFT
      IF(M.EQ.4)GO TO 23                                                DRIFT
      IF(M.NE.3)GO TO 21                                                DRIFT
      CALL EJOINT(-1,J)                                                 DRIFT
      CALL CROSS(HIR(1,2,2*J+29),HIR(1,1,2*J+29),T1)                    DRIFT
      DO 17 I = 1,3                                                     DRIFT
      H1(I) = CONST(4,J)*HIR(I,1,2*J+29) + CONST(5,J)*T1(I)             DRIFT
   17 H2(I) = HIR(I,3,2*J+30)                                           DRIFT
      GO TO 25                                                          DRIFT
   21 DO 22 I = 1,3                                                     DRIFT
      H1(I) = HIR(I,M,2*J+29)                                           DRIFT
   22 H2(I) = HIR(I,M+1,2*J+30)                                         DRIFT
      GO TO 25                                                          DRIFT
   23 DO 24 I = 1,3                                                     DRIFT
      H1(I) = HB(I,2*J-1)                                               DRIFT
```

```
   24 H2(I) = HB(I,2*J)                                                    DRIFT
C                                                                          DRIFT
C      **  ADJUST DC MATRIX FOR CONSTRAINED JOINTS  **                     DRIFT
C                                                                          DRIFT
   25 CALL DOT31(D(1,1,K),H1,T1)                                           DRIFT
      CALL MAT31 (D(1,1,J+1),T1,T2)                                        DRIFT
      CT = T2(1)*H2(1) + T2(2)*H2(2) + T2(3)*H2(3)                         DRIFT
      IF(M.GE.3)GO TO 28                                                   DRIFT
      ST = 1.0/DSQRT((1.0 - CT)*(1.0 + CT))                                DRIFT
      DO 27 I = 1,3                                                        DRIFT
   27 T2(I) = (H2(I) - CT*T2(I))*ST                                        DRIFT
      CT = 1.0/ST                                                          DRIFT
   28 CALL CROSS(H2,T2,T3)                                                 DRIFT
      DO 30 L=1,3                                                          DRIFT
      CALL CROSS (T3,D(1,L,J+1),T4)                                        DRIFT
      ST = T3(1)*D(1,L,J+1) + T3(2)*D(2,L,J+1) + T3(3)*D(3,L,J+1)          DRIFT
      ST = ST/(1.0 + CT)                                                   DRIFT
      DO 30 I=1,3                                                          DRIFT
   30 D(I,L,J+1) = CT*D(I,L,J+1) - T4(I) + ST*T3(I)                        DRIFT
C                                                                          DRIFT
C      **  RENORMALIZATION OF DIRECTION COSINE MATRIX BY **                DRIFT
C      **  AVERAGING MATRIX AND TRANSPOSE OF ITS INVERSE **                DRIFT
C                                                                          DRIFT
      DO 33 ITER= 1,10                                                     DRIFT
      CALL CFACTT (D(1,1,J+1),TP,DET)                                      DRIFT
      DO 32 L = 1,3                                                        DRIFT
      DO 32 I = 1,3                                                        DRIFT
      D(I,L,J+1) = 0.5*(D(I,L,J+1)+TP(L,I)/DET)                            DRIFT
   32 IF (DABS(D(I,L,J+1)).LT.EPS(15)) D(I,L,J+1) = 0.0                    DRIFT
      IF (DABS(DET-1.0).LT.EPS(6)) GO TO 41                                DRIFT
   33 CONTINUE                                                             DRIFT
      WRITE (6,34) J,TIME,DET                                              DRIFT
   34 FORMAT (44H0 DRIFT RENORMALIZATION DID NOT CONVERGE FOR,             DRIFT
     *         10H JOINT NO.,I3,7H TIME =,F10.6,6H DET =,F10.6)            DRIFT
C                                                                          DRIFT
C      **  ADJUST WMEG FOR CONSTRAINED JOINTS  **                          DRIFT
C                                                                          DRIFT
   41 IF(M.NE.4)GO TO 43                                                   DRIFT
      HW = H2(1)*WMEG(1,J+1) - H1(1)*WMEG(1,K)                             DRIFT
     *    + H2(2)*WMEG(2,J+1) - H1(2)*WMEG(2,K)                            DRIFT
     *    + H2(3)*WMEG(3,J+1) - H1(3)*WMEG(3,K)                            DRIFT
      CALL DOT31 (D(1,1,K),WMEG(1,K),T1)                                   DRIFT
      CALL MAT31 (D(1,1,J+1),T1,WMEG(1,J+1))                               DRIFT
      DO 42 I=1,3                                                          DRIFT
   42 WMEG(I,J+1) = WMEG(I,J+1) + HW*H2(I)                                 DRIFT
      GO TO 50                                                             DRIFT
   43 IF(M.NE.3)GO TO 47                                                   DRIFT
      CALL DOT31(D(1,1,K),HIR(1,2,2*J+29),T1)                              DRIFT
      CALL MAT31(D(1,1,J+1),T1,H1)                                         DRIFT
      GO TO 48                                                             DRIFT
```

```
47 CALL MAT31(D(1,1,J+1),T1,T2)                                    DRIFT
   CALL CROSS(T2,H2,H1)                                            DRIFT
48 CALL DOT31(D(1,1,K),WMEG(1,K),T1)                               DRIFT
   CALL MAT31(D(1,1,J+1),T1,T2)                                    DRIFT
   HW = H1(1)*(T2(1) - WMEG(1,J+1))                                DRIFT
  *    + H1(2)*(T2(2) - WMEG(2,J+1))                               DRIFT
  *    + H1(3)*(T2(3) - WMEG(3,J+1))                               DRIFT
   DO 49 I = 1,3                                                   DRIFT
49 WMEG(I,J+1) = WMEG(I,J+1) + HW*H1(I)                            DRIFT
50 CONTINUE                                                        DRIFT
51 RETURN                                                          DRIFT
   END                                                             DRIFT
```

```
      SUBROUTINE DSETD(D,TH,T)                                          DSETD
C                                                REV IV    07/23/86TWOPI
C     UPDATES A DIRECTION COSINE MATRIX (D)                             DSETD
C     USING AN INCREMENTAL ANGULAR MOTION (TH).                         DSETD
C         ARGUMENTS D: 3X3 DIRECTION COSINE MATRIX TO BE UPDATED.       DSETD
C                  TH: 3 COMPONENTS OF INCREMENTAL ANGULAR MOTION       DSETD
C                      ABOUT LOCAL X,Y AND Z AXIS RESPECTIVELY.         DSETD
C                   T: MAGNITUDE OF VECTOR TH COMPUTED BY ROUTINE.      DSETD
C                                                                       DSETD
       IMPLICIT REAL*8(A-H,O-Z)                                         DSETD
      DIMENSION D(3,3),TH(3),S(3),TEMP(3,3)                             DSETD
      COMMON/CNSNTS/ PI,RADIAN,G,THIRD,EPS(24),                         DSETD
     *               UNITL,UNITM,UNITT,GRAVTY(3),TWOPI                  TWOPI
       T=DSQRT(TH(1)**2+TH(2)**2+TH(3)**2)                              DSETD
       IF(T.EQ.0.)RETURN                                                DSETD
       ST=DSIN(T)                                                       DSETD
       CT=DCOS(T)                                                       DSETD
       STT=ST/T                                                         DSETD
       CTT=STT**2/(1.+CT)                                               DSETD
       DO 10 J=1,3                                                      DSETD
       S(1)=-TH(3)*D(2,J)+TH(2)*D(3,J)                                  DSETD
       S(2)= TH(3)*D(1,J)-TH(1)*D(3,J)                                  DSETD
       S(3)=-TH(2)*D(1,J)+TH(1)*D(2,J)                                  DSETD
       DTT=(TH(1)*D(1,J)+TH(2)*D(2,J)+TH(3)*D(3,J))*CTT                 DSETD
       DO 5 K=1,3                                                       DSETD
     5 D(K,J)=D(K,J)*CT-STT*S(K)+TH(K)*DTT                              DSETD
    10 CONTINUE                                                         DSETD
C                                                                       DSETD
C     RENORMALIZATION OF DIRECTION COSINE MATRIX                        DSETD
C     BY AVERAGING MATRIX AND TRANSPOSE OF ITS INVERSE.                 DSETD
C                                                                       DSETD
      DO 23 ITER=1,10                                                   DSETD
      CALL CFACTT(D,TEMP,DET)                                           DSETD
      DO 22 I=1,3                                                       DSETD
      DO 22 J=1,3                                                       DSETD
      D(I,J) = 0.5*(D(I,J)+TEMP(J,I)/DET)                               DSETD
   22 IF (DABS(D(I,J)).LT.EPS(15)) D(I,J)=0.0                           DSETD
      IF (DABS(DET-1.0).LT.EPS(6)) GO TO 24                             DSETD
   23 CONTINUE                                                          DSETD
      WRITE (6,27) DET                                                  DSETD
   27 FORMAT('0 DSETD RENORMALIZATION DID NOT CONVERGE, DET =',1PD25.15)DSETD
   24 RETURN                                                            DSETD
      END                                                               DSETD
```

```
      SUBROUTINE DSETQ(E,TH,ES,EC,D)                                    DSETQ
C                                                  REV IV    07/23/86TWOPI
C     COMPUTES NEW DIRECTION MATRIX (D), GIVEN ORIGINAL MATRIX (E)      DSETQ
C     AND INCREMENTAL MOTION EXPRESSED IN QUATERNION FORM.              DSETQ
C                                                                       DSETQ
C       ARGUMENTS:                                                      DSETQ
C                                                                       DSETQ
C             E : ORIGINAL DIRECTION COSINE MATRIX.                     DSETQ
C            TH : COMPONENTS OF Q ( UX SIN A/2, UY SIN A/2, UZ SIN A/2) DSETQ
C            ES : SIN**2(A/2)                                           DSETQ
C            EC : COS (A/2)                                             DSETQ
C             D : NEW DIRECTION COSINE MATRIX.                          DSETQ
C                                                                       DSETQ
       IMPLICIT REAL*8(A-H,O-Z)                                         DSETQ
      DIMENSION D(3,3),TH(3),S(3),TEMP(3,3),E(3,3)                      DSETQ
      COMMON/CNSNTS/ PI,RADIAN,G,THIRD,EPS(24),                         DSETQ
     *               UNITL,UNITM,UNITT,GRAVTY(3),TWOPI                  TWOPI
      CT = 1.0 - 2.0*ES                                                 DSETQ
      DO 10 J=1,3                                                       DSETQ
      S(1) = TH(2)*E(3,J) - TH(3)*E(2,J)                                DSETQ
      S(2) = TH(3)*E(1,J) - TH(1)*E(3,J)                                DSETQ
      S(3) = TH(1)*E(2,J) - TH(2)*E(1,J)                                DSETQ
      DTT = TH(1)*E(1,J) + TH(2)*E(2,J) + TH(3)*E(3,J)                  DSETQ
      DO 5 K=1,3                                                        DSETQ
    5 D(K,J) = E(K,J)*CT + 2.0*(TH(K)*DTT - EC*S(K))                    DSETQ
   10  CONTINUE                                                         DSETQ
C                                                                       DSETQ
C     RENORMALIZATION OF DIRECTION COSINE MATRIX                        DSETQ
C     BY AVERAGING MATRIX AND TRANSPOSE OF ITS INVERSE.                 DSETQ
C                                                                       DSETQ
      DO 23 ITER=1,10                                                   DSETQ
      CALL CFACTT(D,TEMP,DET)                                           DSETQ
      DO 22 I=1,3                                                       DSETQ
      DO 22 J=1,3                                                       DSETQ
      D(I,J) = 0.5*(D(I,J)+TEMP(J,I)/DET)                               DSETQ
   22 IF (DABS(D(I,J)).LT.EPS(15)) D(I,J)=0.0                           DSETQ
      IF (DABS(DET-1.0).LT.EPS(6)) GO TO 24                             DSETQ
   23 CONTINUE                                                          DSETQ
      WRITE (6,27) DET                                                  DSETQ
   27 FORMAT('0 DSETQ RENORMALIZATION DID NOT CONVERGE, DET =',1PD25.15)DSETQ
   24 RETURN                                                            DSETQ
      END                                                               DSETQ
```

```
      SUBROUTINE DSMSOL (A,KK,LL)                                       DSMSOL
C                                               REV 03    07/08/74DSMSOL
C     SOLVES A SET OF SIMULTANEOUS LINEAR EQUATIONS AX=B.               DSMSOL
C                                                                       DSMSOL
C       ARGUMENTS:                                                      DSMSOL
C         A: 2-DIMENSIONAL(KK,KK+1) MATRIX OF COEFFICIENTS.             DSMSOL
C         KK: NUMBER OF EQUATIONS AND UNKNOWNS.                         DSMSOL
C         LL: 1ST DIMENSION OF A IN CALLING PROGRAM.                    DSMSOL
C                                                                       DSMSOL
C       CALLING PROGRAM SETUP:                                          DSMSOL
C         A(I,J) FOR I,J=1,KK                                           DSMSOL
C         A(I,KK+1) = B(I) FOR I=1,KK                                   DSMSOL
C         THE SOLUTION X IS RETURNED IN COLUMN KK+1 OF A.               DSMSOL
C         MATRIX A IS DESTROYED BY SUBROUTINE.                          DSMSOL
C                                                                       DSMSOL
      IMPLICIT REAL*8(A-H,O-Z)                                          DSMSOL
      DIMENSION A(LL,1)                                                 DSMSOL
      N  = KK                                                           DSMSOL
      N1 = N+1                                                          DSMSOL
      DO 50  L=1,N                                                      DSMSOL
      L1 = L+1                                                          DSMSOL
      BIG = 0.0                                                         DSMSOL
      DO 25  I=L,N                                                      DSMSOL
      IF (DABS(A(I,L)).LE.DABS(BIG)) GO TO 25                           DSMSOL
      K = I                                                             DSMSOL
      BIG = A(I,L)                                                      DSMSOL
   25 CONTINUE                                                          DSMSOL
      IF (BIG.NE.0.0) GO TO 30                                          DSMSOL
      WRITE (6,26)                                                      DSMSOL
   26 FORMAT('0 DSMSOL MATRIX SINGULAR, PROGRAM TERMINATED.')           DSMSOL
      STOP  41                                                          DSMSOL
   30 BIG = 1.0/BIG                                                     DSMSOL
      DO 40  J=L,N1                                                     DSMSOL
      B      = A(K,J)                                                   DSMSOL
      A(K,J) = A(L,J)                                                   DSMSOL
   40 A(L,J) = B*BIG                                                    DSMSOL
      IF (L.EQ.N) GO TO 50                                              DSMSOL
      DO 48  I=L1,N                                                     DSMSOL
      IF (A(I,L).EQ.0.0) GO TO 48                                       DSMSOL
      DO 45  J=L1,N1                                                    DSMSOL
   45 A(I,J) = A(I,J)-A(I,L)*A(L,J)                                     DSMSOL
   48 CONTINUE                                                          DSMSOL
   50 CONTINUE                                                          DSMSOL
      IF (N.EQ.1) GO TO 71                                              DSMSOL
      N2 = N-1                                                          DSMSOL
      DO 60  L=1,N2                                                     DSMSOL
      I  = N-L                                                          DSMSOL
      L1 = I+1                                                          DSMSOL
      DO 60  J=L1,N                                                     DSMSOL
   60 A(I,N1) = A(I,N1)-A(I,J)*A(J,N1)                                  DSMSOL
```

```
   71 CONTINUE                                                        DSMSOL
      RETURN                                                          DSMSOL
      END                                                             DSMSOL
```

```
      SUBROUTINE DZP(N,X,GG,E,R,M)                                      DZP
C                                                   REV IV    07/23/86TWOPI
C     COMPUTES THE STATE VARIABLES (X) FROM THE PARAMETRIC FORM ASSUMED DZP
C     IN THE INTEGRATION ROUTINE DINT. ALSO EVALUATES THE EXPONENTIAL   DZP
C     WEIGHTS (E) IF M IS NOT ZERO.                                     DZP
C                                                                       DZP
      IMPLICIT REAL*8 (A-H,O-Z)                                         DZP
      DIMENSION X(1),GG(5,1),E(3,1)                                     DZP
      COMMON/CNSNTS/ PI,RADIAN,G,THIRD,EPS(24),                         DZP
     *               UNITL,UNITM,UNITT,GRAVTY(3),TWOPI                  TWOPI
C                                                                       DZP
      CALL ELTIME(1,5)                                                  DZP
      IF(M.NE.0) GO TO 10                                               DZP
C                                                                       DZP
C     COMPUTE STATE VARIABLES ONLY.                                     DZP
C                                                                       DZP
      DO 5  I=1,N                                                       DZP
    5  X(I) = GG(1,I) + R*(GG(2,I)*E(1,I)                               DZP
     *                + R*(GG(3,I)*E(2,I)                               DZP
     *                + R*(GG(4,I)*E(3,I) )))                           DZP
      GO TO 90                                                          DZP
C                                                                       DZP
C     COMPUTE EXPONENTIAL WEIGHTS AND STATE VARIABLES.                  DZP
C                                                                       DZP
   10 DO 50  I=1,N                                                      DZP
      E(1,I) = 1.0                                                      DZP
      E(2,I) = 0.5                                                      DZP
      E(3,I) = THIRD                                                    DZP
      IF (GG(5,I).EQ.0.0) GO TO 50                                      DZP
      Z = R*GG(5,I)                                                     DZP
      W = 0.                                                            DZP
      IF (DABS(Z).GT.0.004) GO TO 20                                    DZP
      W = 4.                                                            DZP
      A = E(3,I)                                                        DZP
      E(3,I) = 0.                                                       DZP
   15 E(3,I) = E(3,I)+A                                                 DZP
      A = A*Z/W                                                         DZP
      W = W+1.0                                                         DZP
      IF(E(3,I)+A.NE.E(3,I)) GO TO 15                                   DZP
      E(2,I) = 0.5+0.5*Z*E(3,I)                                         DZP
      E(1,I) = 1.+Z*E(2,I)                                              DZP
      GO TO 50                                                          DZP
   20 IF(Z.GT.-40.) W = DEXP(Z)                                         DZP
      E(1,I) = (W-1.)/Z                                                 DZP
      E(2,I) = (E(1,I)-1.)/Z                                            DZP
      E(3,I) = (2.*E(2,I)-1.)/Z                                         DZP
   50  X(I) = GG(1,I) + R*(GG(2,I)*E(1,I)                               DZP
     *                + R*(GG(3,I)*E(2,I)                               DZP
     *                + R*(GG(4,I)*E(3,I) )))                           DZP
C                                                                       DZP
```

```
90 CALL ELTIME(2,5)                                                      DZP
   RETURN                                                                DZP
   END                                                                   DZP
```

```
      SUBROUTINE EDEPTH (A,B,XM,T,Y,XA,XB,XL,XU)                        EDEPTH
                                                    REV IV    07/23/86TWOPI
C                                                                       EDEPTH
C     DETERMINES XA AND XB, THE POINTS OF MAXIMUM PENETRATION OF TWO    EDEPTH
C     INTERSECTING ELLIPSOIDS A AND B.                                  EDEPTH
C      ARGUMENTS A,B,XM,T AND X SAME AS FOR SUBROUTINE INTERS.          EDEPTH
C     ARGUMENTS XL AND XU, IF NONZERO, ARE FINAL RESULTS OF LAST CALL.  EDEPTH
C                                                                       EDEPTH
      IMPLICIT REAL*8 (A-H,O-Z)                                         EDEPTH
      DIMENSION A(3,3),B(3,3),XM(3),Y(3),XA(3),XB(3)                    EDEPTH
      COMMON/CONTRL/ TIME,NSEG,NJNT,NPL,NBLT,NBAG,NVEH,NGRND,           EDEPTH
     *               NS,NQ,NSD,NFLX,NHRNSS,NWINDF,NJNTF,NPRT(36),NPG    PAGE
      COMMON/CNSNTS/ PI,RADIAN,G,THIRD,EPS(24),                         EDEPTH
     *               UNITL,UNITM,UNITT,GRAVTY(3),TWOPI                  TWOPI
      DIMENSION C1(3,4),C2(3,4),C3(3,4),XBM(3),PXBL(3),PXAU(3),AB(3,3)  EDEPTH
      DIMENSION AXA(3),BXBM(3),PXAL(3),PXBU(3)                          EDEPTH
      EQUIVALENCE (XBM(1),C1(1,4)), (PXBL(1),C2(1,4)), (PXAU(1),C3(1,4))EDEPTH
C                                                                       EDEPTH
C                                   INITIAL GUESSES                     EDEPTH
C                                   XA = Y/T                            EDEPTH
C                                   XB = M+(Y-M)/T                      EDEPTH
C                                   L = -!XB-XA!/!AXA!                  EDEPTH
C                                   U = -!XB-XA!/!B(XB-M)!              EDEPTH
C                                                                       EDEPTH
      D1 = 0.0                                                          EDEPTH
      D2 = 0.0                                                          EDEPTH
      DO  9 I=1,3                                                       EDEPTH
      XA(I) = Y(I)/T                                                    EDEPTH
      XBM(I) = (Y(I)-XM(I))/T                                           EDEPTH
      XB (I) = XBM(I)+XM(I)                                             EDEPTH
    9 D1 = D1+(XB(I)-XA(I))**2                                          EDEPTH
      IF (DABS(T-1.0).LE.EPS(6)) GO TO 31                               EDEPTH
      ITER = 0                                                          EDEPTH
      CALL MAT33 (A,B,AB)                                               EDEPTH
      IF  (XL.NE.0.0)  GO TO 11                                         EDEPTH
      IF  (XU.NE.0.0)  GO TO 11                                         EDEPTH
      D3 = 0.0                                                          EDEPTH
      DO 10 I=1,3                                                       EDEPTH
      AXA(I) = A(I,1)*XA(1)                                             EDEPTH
     *       + A(I,2)*XA(2)                                             EDEPTH
     *       + A(I,3)*XA(3)                                             EDEPTH
      D2 = D2 + AXA(I)**2                                               EDEPTH
      BXBM(I) = B(I,1)*XBM(1)                                           EDEPTH
     *        + B(I,2)*XBM(2)                                           EDEPTH
     *        + B(I,3)*XBM(3)                                           EDEPTH
   10 D3 = D3+BXBM(I)**2                                                EDEPTH
      XL = -DSQRT(D1/D2)                                                EDEPTH
      XU = -DSQRT(D1/D3)                                                EDEPTH
C                                                                       EDEPTH
C                                   START OF ITERATION                  EDEPTH
C                                                                       EDEPTH
```

```
   11 ITER = ITER+1                                                   EDEPTH
      IF (NPRT(17).NE.0) WRITE (6,12) ITER,XL,XU,XA,XB                EDEPTH
   12 FORMAT(' EDEPTH ITER',I6,8G14.6)                                EDEPTH
      IF (ITER.LE.50) GO TO 14                                        EDEPTH
      WRITE (6,13)                                                    EDEPTH
   13 FORMAT(' EDEPTH ITERATION DID NOT CONVERGE')                    EDEPTH
      GO TO 31                                                        EDEPTH
C                                                                     EDEPTH
C                                  FORM MATRICES                      EDEPTH
C                                  C1 = LUAB + LA + UB                EDEPTH
C                                  C2 = C1                            EDEPTH
C                                  C3 = C1'                           EDEPTH
C                                                                     EDEPTH
   14 XLAU = XU*XL                                                    EDEPTH
      DO 22 I=1,3                                                     EDEPTH
      XBM(I) = 0.0                                                    EDEPTH
      DO 22 J=1,3                                                     EDEPTH
      C1(I,J) = XLAU*AB(I,J) + XL*A(I,J) + XU*B(I,J)                  EDEPTH
      C2(I,J) = C1(I,J)                                               EDEPTH
      C3(J,I) = C1(I,J)                                               EDEPTH
   22 XBM(I) = XBM(I) - XL*A(I,J)*XM(J)                               EDEPTH
C                                                                     EDEPTH
C                                  SOLVE FOR (XB-M)                   EDEPTH
C                                  C1(XB-M) = -LAM                    EDEPTH
C                                                                     EDEPTH
      CALL DSMSOL(C1,3,3)                                             EDEPTH
C                                                                     EDEPTH
C                                  EVALUATE                           EDEPTH
C                                  XB = (XB-M)+M                      EDEPTH
C                                  B(XB-M)                            EDEPTH
C                                  AXA                                EDEPTH
C                                  C13 = (1-XA'AXA)/2                 EDEPTH
C                                  C23 = (1-(XB-M)'B(XB-M))/2         EDEPTH
C                                                                     EDEPTH
      C13 = 0.0                                                       EDEPTH
      C23 = 0.0                                                       EDEPTH
      DO 23 I=1,3                                                     EDEPTH
      XB(I) = XBM(I)+XM(I)                                            EDEPTH
      BXBM(I) = B(I,1)*XBM(1)                                         EDEPTH
     *        + B(I,2)*XBM(2)                                         EDEPTH
     *        + B(I,3)*XBM(3)                                         EDEPTH
   23 XA(I) = XB(I) + XU*BXBM(I)                                      EDEPTH
      DO 24 I=1,3                                                     EDEPTH
      AXA(I) = A(I,1)*XA(1)                                           EDEPTH
     *       + A(I,2)*XA(2)                                           EDEPTH
     *       + A(I,3)*XA(3)                                           EDEPTH
      C13 = C13 + XA(I)*AXA(I)                                        EDEPTH
      C23 = C23 + XBM(I)*BXBM(I)                                      EDEPTH
   24 PXBL(I) = -AXA(I)                                               EDEPTH
      C13 = (1.0-C13)/2.0                                             EDEPTH
```

```
      C23 = (1.0-C23)/2.0                                                 EDEPTH
C                                      *         DXB                        EDEPTH
C                                      SOLVE FOR ---                        EDEPTH
C                                      *         DL                         EDEPTH
C                                                                           EDEPTH
C                                      * DXB                                EDEPTH
C                                      C2--- = -AXA                         EDEPTH
C                                      * DL                                 EDEPTH
C                                                                           EDEPTH
      CALL DSMSOL(C2,3,3)                                                 EDEPTH
C                                                                           EDEPTH
C                                      CALCULATE                            EDEPTH
C                                      DXA   DXB     DXB                    EDEPTH
C                                      --- = --- + UB---                    EDEPTH
C                                      DL    DL      DL                     EDEPTH
C                                                                           EDEPTH
C                                      *         DXA                        EDEPTH
C                                      C11 = XA'A---                        EDEPTH
C                                      *         DL                         EDEPTH
C                                                                           EDEPTH
C                                      *             DXB                    EDEPTH
C                                      C21 = (XB-M)'B---                    EDEPTH
C                                      *             DL                     EDEPTH
C                                                                           EDEPTH
      C11 = 0.0                                                           EDEPTH
      C21 = 0.0                                                           EDEPTH
      DO 25 I=1,3                                                         EDEPTH
      PXAL(I) = B(I,1)*PXBL(1)                                            EDEPTH
     *         + B(I,2)*PXBL(2)                                           EDEPTH
     *         + B(I,3)*PXBL(3)                                           EDEPTH
      PXAL(I) = PXBL(I) + XU*PXAL(I)                                      EDEPTH
      C11 = C11 + AXA(I)*PXAL(I)                                          EDEPTH
      C21 = C21 + BXBM(I)*PXBL(I)                                         EDEPTH
   25 PXAU(I) = -BXBM(I)                                                  EDEPTH
C                                                                           EDEPTH
C                                      *         DXA                        EDEPTH
C                                      SOLVE FOR ---                        EDEPTH
C                                      *         DU                         EDEPTH
C                                                                           EDEPTH
C                                      * DXA                                EDEPTH
C                                      C3--- = -B(XB-M)                     EDEPTH
C                                      * DU                                 EDEPTH
C                                                                           EDEPTH
      CALL DSMSOL(C3,3,3)                                                 EDEPTH
C                                                                           EDEPTH
C                                      CALCULATE                            EDEPTH
C                                      DXB   DXA     DXA                    EDEPTH
C                                      --- = --- + LA---                    EDEPTH
C                                      DU    DU      DU                     EDEPTH
C                                                                           EDEPTH
```

```
C                                          *        DXA                        EDEPTH
C                                          C12 = XA'A---                       EDEPTH
C                                          *        DU                         EDEPTH
C                                                                              EDEPTH
C                                          *            DXB                    EDEPTH
C                                          C22 = (XB-M)'B---                   EDEPTH
C                                          *            DU                     EDEPTH
C                                                                              EDEPTH
      C12 = 0.0                                                                EDEPTH
      C22 = 0.0                                                                EDEPTH
      DO 26 I=1,3                                                              EDEPTH
      PXBU(I) = PXAU(I) + XL*(A(I,1)*PXAU(1)                                   EDEPTH
     *     + A(I,2)*PXAU(2) + A(I,3)*PXAU(3) )                                 EDEPTH
      C12 = C12 + AXA(I)*PXAU(I)                                               EDEPTH
   26 C22 = C22 + BXBM(I)*PXBU(I)                                              EDEPTH
C                                                                              EDEPTH
C                                          SOLVE FOR DL AND DU                 EDEPTH
C                                          C11*DL + C12*DU = C13               EDEPTH
C                                          C21*DL + C22*DU = C23               EDEPTH
C                                                                              EDEPTH
      DET = C11*C22-C12*C21                                                    EDEPTH
      DL = (C13*C22-C12*C23)/DET                                               EDEPTH
      DU = (C11*C23-C13*C21)/DET                                               EDEPTH
C                                                                              EDEPTH
C                                          INCREMENT L AND U                   EDEPTH
C                                          TEST FOR CONVERGENCE                EDEPTH
C                                                                              EDEPTH
      XL = XL + DL                                                             EDEPTH
      XU = XU + DU                                                             EDEPTH
      IF (DABS(DL/XL).GT.EPS(12)) GO TO 11                                     EDEPTH
      IF (DABS(DU/XU).GT.EPS(12)) GO TO 11                                     EDEPTH
   31 CONTINUE                                                                 EDEPTH
      RETURN                                                                   EDEPTH
      END                                                                      EDEPTH
```

```
      DOUBLE PRECISION FUNCTION EFUNCT (TH,THD,SPR,JSTOP)                EFUNCT
C                                                    REV 20    04/29/80EFUNCT
C     COMPUTES NONLINEAR SRRING TORQUE FOR EULER JOINTS.                 EFUNCT
C                                                                        EFUNCT
C     ARGUMENTS:                                                         EFUNCT
C          TH    - THETA IS THE ANGLE OF THE EULER AXIS                  EFUNCT
C          THD   - THETA DOT                                             EFUNCT
C          SPR   - ARRAY OF 5 VALUES DESCRIBING FUNCTION EVALUATION      EFUNCT
C          JSTOP - INDICATOR TO BE SET TO ONE IF IN STOP                 EFUNCT
C                                                                        EFUNCT
      IMPLICIT REAL*8(A-H,O-Z)                                           EFUNCT
      DIMENSION SPR(5)                                                   EFUNCT
      JSTOP = 0                                                          EFUNCT
      EFUNCT = TH*SPR(1)                                                 EFUNCT
      TEN = 10.0                                                         EFUNCT
      Q = DSIGN(TEN*THD,TH*THD)                                          EFUNCT
      IF (Q.GT.1.0) Q = 1.0                                              EFUNCT
      IF (Q.LT.-1.0) Q = -1.0                                            EFUNCT
      X = 0.5*(1.0+SPR(4)+Q*(1.0-SPR(4)))                                EFUNCT
      IF (SPR(5).GT.0.0) GO TO 10                                        EFUNCT
      EFUNCT = X*EFUNCT                                                  EFUNCT
      GO TO 99                                                           EFUNCT
   10 IF (DABS(TH).LT.SPR(5)) GO TO 99                                   EFUNCT
      JSTOP = 1                                                          EFUNCT
      Z = DABS(TH) - SPR(5)                                              EFUNCT
      EFUNCT = EFUNCT + DSIGN(X*(SPR(2)+Z*SPR(3))*Z**2,TH)               EFUNCT
   99 RETURN                                                             EFUNCT
      END                                                                EFUNCT
```

```
      SUBROUTINE EJOINT(IJ,NK)                                          JDRIFT
C                                                 REV IV    07/24/86SLIP
C     COMPUTES THE TORQUES ACTING ON AN EULER JOINT                     EJOINT
C     AND ADDS THEM TO THE U2 ARRAY.                                    EJOINT
C                                                                       EJOINT
C     ARGUMENTS:                                                        EJOINT
C        NK = 0 - REGULAR COMPUTATION FOR ALL EULER JOINTS              JDRIFT
C           # 0 - COMPUTE ONLY FOR JOINT NJ IMPULSE                     EJOINT
C                                                                       EJOINT
C        IJ = 1 IMPULSE ON PRECESSION AXIS ONLY                         EJOINT
C           = 2 IMPULSE ON NUTATION AXIS ONLY                           EJOINT
C           = 3 IMPULSE ON SPIN AXIS ONLY                               EJOINT
C           = 4 IMPULSE ON GLOBALGRAPHIC AXIS                           EJOINT
C        NK = 0, IJ # 0, SPECIAL COMPUTATIONS OF HIR AND HB ONLY        JDRIFT
C                                                                       EJOINT
      IMPLICIT REAL*8(A-H,O-Z)                                          EJOINT
      COMMON/CONTRL/ TIME,NSEG,NJNT,NPL,NBLT,NBAG,NVEH,NGRND,           EJOINT
     *               NS,NQ,NSD,NFLX,NHRNSS,NWINDF,NJNTF,NPRT(36),NPG    PAGE
      COMMON/SGMNTS/ D(3,3,30),WMEG(3,30),WMEGD(3,30),U1(3,30),U2(3,30),EJOINT
     *               SEGLP(3,30),SEGLV(3,30),SEGLA(3,30),NSYM(30)       EJOINT
      COMMON/DESCRP/ PHI(3,30),W(30),RW(30),SR(4,60),HA(3,60),HB(3,60), SLIP
     *               RPHI(3,30),HT(3,3,60),SPRING(5,90),VISC(7,90),     EJOINT
     *               JNT(30),IPIN(30),ISING(30),IGLOB(30),JOINTF(30)    EJOINT
      COMMON/CMATRX/ V1(3,30),V2(3,30),V3(3,12),B12(3,3,60),A22(3,3,60),EJOINT
     *               F(3,30),TQ(3,30),WJ(30),A11(3,3,30)                SLIP
      COMMON/CEULER/ IEULER(30),HIR(3,3,90),ANG(3,30),ANGD(3,30),       JDRIFT
     *               FE(3,30),TQE(3,30),CONST(5,30)                     JDRIFT
      COMMON/FORCES/PSF(7,70),BSF(4,20),SSF(10,40),BAGSF(3,20),         NCFORC
     *               PRJNT(7,30),NPANEL(5),NPSF,NBSF,NSSF,NBGSF         EJOINT
      COMMON/TEMPVI/ CREST,TTI(3),R1I(3),R2I(3),JSTOP(4,2,30)           EJOINT
      COMMON/CNSNTS/ PI,RADIAN,G,THIRD,EPS(24),                         EJOINT
     *               UNITL,UNITM,UNITT,GRAVTY(3),TWOPI                  TWOPI
      COMMON/TEMPVS/ DH1(3,3),DH4(3,3),TH(3,3),HIM(3,3),HIJ(3,3),       EJOINT
     *               HDT(3,3),H2(3,3),SH(3),TM(3),TJ(3),WMJ(3),AD(3),   EJOINT
     *               CV(3),CS(3),ANGL(3),HD3(3),CC(3),T9(3),LSKIP(3)    EJOINT
      LOGICAL LSKIP                                                     EJOINT
      IF (NJNT.LE.0) GO TO 99                                           EJOINT
      CALL ELTIME(1,31)                                                 EJOINT
      J1 = 1                                                            EJOINT
      J2 = NJNT                                                         EJOINT
      NJ = NK                                                           JDRIFT
      IF (NJ.EQ.0) GO TO 11                                             EJOINT
      J1 = NJ                                                           EJOINT
      J2 = NJ                                                           EJOINT
      IF(IJ.LT.0) NJ = 0                                                JDRIFT
   11 DO 98 J=J1,J2                                                     EJOINT
      IF (IABS(IPIN(J)).NE.4) GO TO 98                                  EJOINT
      M = IABS(JNT(J))                                                  EJOINT
      CALL DOT33(D(1,1,M),HT(1,1,2*J-1),DH1)                            EJOINT
      CALL DOT33(D(1,1,J+1),HT(1,1,2*J),DH4)                            EJOINT
```

```
      CALL DOT33(DH4,DH1,TH)                                     EJOINT
      DO 12 I=1,3                                                EJOINT
   12 ANG(I,J) = ANG(I,J) + CONST(I,J)                           EJOINT
      IC = IEULER(J)                                             EJOINT
      CALL EULRAD (TH,ANG(1,J),IC)                               EJOINT
      CALL ROT(H2,3,-ANG(1,J))                                   EJOINT
      DO 13 I=1,3                                                EJOINT
      ANG(I,J) = ANG(I,J) - CONST(I,J)                           EJOINT
      HIR(I,1,J) = DH1(I,3)                                      EJOINT
      HIR(I,3,J) = DH4(I,3)                                      EJOINT
      HIM(I,1) = HT(I,3,2*J-1)                                   EJOINT
      HIJ(I,3) = HT(I,3,2*J)                                     EJOINT
      LSKIP(I) = .FALSE.                                         EJOINT
      FE(I,J) = 0.0                                              EJOINT
      CV(I)   = 0.0                                              EJOINT
      CS(I)   = 0.0                                              EJOINT
      V2(I,J) = 0.0                                              EJOINT
      TQE(I,J) = 0.0                                             EJOINT
   13 TQ(I,J) = 0.0                                              EJOINT
      WJ(J)   = 0.0                                              EJOINT
      TQC = 0.0                                                  EJOINT
      IF (IJ.EQ.4) GO TO 55                                      EJOINT
      CALL MAT31 (HT(1,1,2*J-1),H2(1,1),HIM(1,2))                EJOINT
      CALL MAT31 (HT(1,1,2*J-1),H2(1,2),HIM(1,3))                EJOINT
      CALL DOT31 (D(1,1,M),HIM(1,2),H2(1,2))                     EJOINT
      CALL DOT31 (D(1,1,M),HIM(1,3),H2(1,3))                     EJOINT
      CALL CROSS (H2(1,2),HIR(1,3,J),H2(1,1))                    EJOINT
      CALL DOT31 (D(1,1,M  ),WMEG(1,M  ),TM)                     EJOINT
      CALL DOT31 (D(1,1,J+1),WMEG(1,J+1),TJ)                     EJOINT
      SWJ = 0.0                                                  EJOINT
      DO 14 I=1,3                                                EJOINT
      HIR(I,2,J) = H2(I,2)                                       EJOINT
      WMJ(I) = TJ(I) - TM(I)                                     EJOINT
   14 SWJ = SWJ + WMJ(I)**2                                      EJOINT
      WJ(J) = DSQRT(SWJ)                                         EJOINT
      CALL DOT31 (HIR(1,1,J),WMJ,AD)                             EJOINT
      CALL CROSS (TM,HIR(1,1,J),HDT(1,1))                        EJOINT
      CALL CROSS (TM,HIR(1,2,J),HDT(1,2))                        EJOINT
      CALL CROSS (TJ,HIR(1,3,J),HDT(1,3))                        EJOINT
      CALL MAT31 (D(1,1,J+1),HIR(1,1,J),HIJ(1,1))                EJOINT
      CALL MAT31 (D(1,1,J+1),HIR(1,2,J),HIJ(1,2))                EJOINT
      CALL MAT31 (D(1,1,M  ),HIR(1,3,J),HIM(1,3))                EJOINT
      N = IEULER(J)                                              EJOINT
      DO 15 I=1,3                                                EJOINT
      SH(I) = AD(I)                                              JDRIFT
      DO 15 K=1,3                                                JDRIFT
      HIR(I,K,2*J+29) = HIM(I,K)                                 JDRIFT
   15 HIR(I,K,2*J+30) = HIJ(I,K)                                 JDRIFT
      IF (N.EQ.8) GO TO 19                                       EJOINT
      IF (N.GT.3) GO TO 16                                       EJOINT
```

```
      SH(N) = 0.0                                                          EJOINT
      GO TO 18                                                             EJOINT
   16 DO 17 I=1,3                                                          EJOINT
   17 IF (I.NE.N-3) SH(I) = 0.0                                            EJOINT
   18 IF (N.NE.2) GO TO 21                                                 EJOINT
   19 HX = H2(1,1)*HIR(1,1,J) + H2(2,1)*HIR(2,1,J) + H2(3,1)*HIR(3,1,J) EJOINT
      IF (DABS(HX).GE.EPS(6)) GO TO 20                                     EJOINT
      SH(1) = ANGD(1,J)                                                    EJOINT
      SH(3) = ANGD(3,J)                                                    EJOINT
      GO TO 21                                                             EJOINT
   20 CALL DOT31 (H2,WMJ,SH)                                               EJOINT
      SH(1) = SH(1)/HX                                                     EJOINT
      IF  (N.EQ.2)  SH(2) = 0.0                                            EJOINT
      SH(3) = SH(3)/HX                                                     EJOINT
   21 DO 22 I=1,3                                                          EJOINT
      ANGD(I,J) = SH(I)                                                    EJOINT
   22 HDT(I,2) = HDT(I,2) + SH(1)*H2(I,3)                                  EJOINT
      IF (NJ.NE.0) N = IJ+3                                                EJOINT
      IF (N.GT.3) GO TO 30                                                 EJOINT
      N4 = 4-N                                                             EJOINT
      IF (N.EQ.2) AHDT = HDT(1,2)*WMJ(1)+HDT(2,2)*WMJ(2)+HDT(3,2)*WMJ(3)EJOINT
      IF (N.NE.2) AHDT = -(SH(2)*HDT(1,2)+SH(N4)*HDT(1,N4))*H2(1,N)        EJOINT
     *                   -(SH(2)*HDT(2,2)+SH(N4)*HDT(2,N4))*H2(2,N)        EJOINT
     *                   -(SH(2)*HDT(3,2)+SH(N4)*HDT(3,N4))*H2(3,N)        EJOINT
      CALL MAT31 (D(1,1,M  ),H2(1,N),HB(1,2*J-1))                          EJOINT
      CALL MAT31 (D(1,1,J+1),H2(1,N),HB(1,2*J  ))                          EJOINT
      DO 25 I=1,3                                                          EJOINT
      V2(I,J) = AHDT*H2(I,N)                                               EJOINT
   25 IF (N.EQ.I) LSKIP(I) = .TRUE.                                        EJOINT
      GO TO 42                                                             EJOINT
   30 IF (N.GT.6) GO TO 40                                                 EJOINT
      K3J = 3*J-2                                                          EJOINT
      DO 32 I=1,3                                                          EJOINT
      IF (NJ.EQ.0) GO TO 31                                                EJOINT
      IF  (I.EQ.N-3)  CREST = VISC(7,K3J)                                  EJOINT
      TQE(I,J) = H2(I,N-3)                                                 EJOINT
      GO TO 32                                                             EJOINT
   31 V2(I,J) = -HDT(I,N-3)*AD(N-3)                                        EJOINT
      HB(I,2*J-1) = HIM(I,N-3)                                             EJOINT
      HB(I,2*J  ) = HIJ(I,N-3)                                             EJOINT
      IF (I.NE.N-3) LSKIP(I) = .TRUE.                                      EJOINT
   32 K3J = K3J + 1                                                        EJOINT
      IF  (NJ)  35,42,35                                                   EJOINT
   40 IF (N.EQ.7) GO TO 97                                                 EJOINT
   42 IF(IJ.NE.0) GOTO 98                                                  JDRIFT
      DO 41 I=1,3                                                          JDRIFT
      IF  (LSKIP(I))  GO TO 41                                             EJOINT
      K3J = 3*J-3+I                                                        EJOINT
      CV(I) = ANGD(I,J)*VISCOS(DABS(ANGD(I,J)),VISC(1,K3J),HA(I,2*J))      EJOINT
      CS(I) = EFUNCT(ANG(I,J),ANGD(I,J),SPRING(1,K3J),JSTOP(I,1,J))        EJOINT
```

```
      FE(I,J) = CS(I) + CV(I) + HA(I,2*J)*HA(I,2*J-1)                   EJOINT
   41 CONTINUE                                                          EJOINT
      CALL MAT31(HIR(1,1,J),FE(1,J),TQE(1,J))                           EJOINT
      IF(NJ.GT.0) GO TO 34                                              EJOINT
   55 IF (IGLOB(J).EQ.0) GO TO 34                                       EJOINT
      HD3(1) = TH(3,1)                                                  EJOINT
      HD3(2) = TH(3,2)                                                  EJOINT
      HD3(3) = TH(3,3)                                                  EJOINT
      CALL GLOBAL (J,HD3,DH1,TQC,T9,ANGL)                               EJOINT
   34 CONTINUE                                                          EJOINT
C                                                                       EJOINT
C     ADD TORQUE CONVERTED TO LOCAL REFERENCE TO U2 ARRAY BY            EJOINT
C      U2(M  ) = U2(M  ) + D(M  )*TQ                                    EJOINT
C      U2(J+1) = U2(J+1) - D(J+1)*TQ                                    EJOINT
C                                                                       EJOINT
   35 DO  51  I=1,3                                                     EJOINT
      TQ(I,J) = TQE(I,J)+TQC*T9(I)                                      EJOINT
      TTI(I) = TQ(I,J)                                                  EJOINT
      DO 51 K=1,3                                                       EJOINT
      U2(K,M  ) = U2(K,M  ) + D(K,I,M  )*TQ(I,J)                        EJOINT
   51 U2(K,J+1) = U2(K,J+1) - D(K,I,J+1)*TQ(I,J)                        EJOINT
C                                                                       EJOINT
C     STORE DATA INTO PRJNT ARRAY FOR OUTPUT ROUTINE                    EJOINT
C                                                                       EJOINT
   97 PRJNT(1,J) = IEULER(J)                                            EJOINT
      PRJNT(2,J) = ANG(1,J)                                             EJOINT
      PRJNT(3,J) = ANG(2,J)                                             EJOINT
      PRJNT(4,J) = ANG(3,J)                                             EJOINT
      PRJNT(5,J)=CS(1)**2+CS(3)**2+2.0*CS(1)*CS(3)*TH(3,3)+CS(2)**2     JTF785
      PRJNT(6,J)=CV(1)**2+CV(3)**2+2.0*CV(1)*CV(3)*TH(3,3)+CV(2)**2     JTF785
      PRJNT(7,J) = TQ(1,J)**2 + TQ(2,J)**2 + TQ(3,J)**2                 EJOINT
   98 CONTINUE                                                          EJOINT
      CALL ELTIME(2,31)                                                 EJOINT
   99 RETURN                                                            EJOINT
      END                                                               EJOINT
```

```
      DOUBLE PRECISION FUNCTION ELONG(A,B,C,D,E)                         ELONG
C                                                 REV 01    10/05/72ELONG
C     COMPUTES ARC LENGTH OF ELLIPSE       AX**2 + 2BXY + CY**2 = 1     ELONG
C     FROM THETA=0 (POSITIVE X AXIS) TO THETA=E (RADIANS)               ELONG
C     WHERE D IS NOMINAL INCREMENT OF INTEGRATION.                      ELONG
C                                                                       ELONG
      IMPLICIT REAL*8(A-H,O-Z)                                          ELONG
      N=DABS(E/D)                                                       ELONG
      N=N+N                                                             ELONG
      IF(N.EQ.0)N=2                                                     ELONG
      Z=N                                                               ELONG
      T=E/Z                                                             ELONG
      F = DSQRT ((1.+(B/A)**2)/A)                                       ELONG
      CS=1.                                                             ELONG
      SN=0.                                                             ELONG
      DCS=DCOS(T)                                                       ELONG
      DSN=DSIN(T)                                                       ELONG
      S=F/2.                                                            ELONG
      AC = A+C                                                          ELONG
      BAC = B*B-A*C                                                     ELONG
      DO 10 I=1,N,2                                                     ELONG
      CSS=CS*DCS-SN*DSN                                                 ELONG
      SN=SN*DCS+CS*DSN                                                  ELONG
      CS=CSS                                                            ELONG
      G=(A*CS+B*SN)*CS+(B*CS+C*SN)*SN                                   ELONG
      G = G**2/(AC + BAC/G)                                             ELONG
      F=(F+1./(F*G))/2.                                                 ELONG
      S=S+F+F                                                           ELONG
      CSS=CS*DCS-SN*DSN                                                 ELONG
      SN=SN*DCS+CS*DSN                                                  ELONG
      CS=CSS                                                            ELONG
      G=(A*CS+B*SN)*CS+(B*CS+C*SN)*SN                                   ELONG
      G = G**2/(AC + BAC/G)                                             ELONG
      F=(F+1./(F*G))/2.                                                 ELONG
      S=S+F                                                             ELONG
   10 CONTINUE                                                          ELONG
      ELONG=(S+S-F)*T/3.                                                ELONG
      RETURN                                                            ELONG
      END                                                               ELONG
```

```
      SUBROUTINE ELTIME(L,N)                                             ELTIME
C                                                 REV III.2 08/08/84REVIII
C     COUNTS THE NUMBER OF TIMES CERTAIN BASIC SUBROUTINES ARE CALLED   ELTIME
C     AND ACCOUNTS FOR ALL COMPUTER CPU TIME USED BY THESE ROUTINES.    ELTIME
C                                                                        ELTIME
C       ARGUMENTS L: 1 INDICATES CALL IS AT START OF ROUTINE            ELTIME
C                    2 INDICATES CALL IS AT END OF ROUTINE.             ELTIME
C                   >2 PAGE NUMBER FOR CALL AT END OF RUN               PAGE
C                 N: THE SUBROUTINE IDENTIFICATION NUMBER.              ELTIME
C                                                                        ELTIME
C       ASSUMES FUNCTION LTIME(1) IS GIVING ELAPSED CPU TIME IN INTEGER ELTIME
C       UNITS OF 0.01 SECONDS SINCE FUNCTION LTIME(0) WAS CALLED.       ELTIME
C                                                                        ELTIME
      DIMENSION NT(40),MTIN(40),NC(40),IND(40)                           ELTIME
      REAL*8 SUB(40)                                                     ELTIME
      DATA   SUB/                                                        ELTIME
     *   8H MAIN3D ,8H INPUT  ,8H DINT   ,8H PRIPLT ,8H DZP    ,         ELTIME
     *   8H PDAUX  ,8H UPDATE ,8H OUTPUT ,8H DAUX   ,8H SETUP1 ,         ELTIME
     *   8H CHAIN  ,8H CONTCT ,8H VISPR  ,8H DAUX11 ,8H DAUX12 ,         ELTIME
     *   8H DAUX22 ,8H DAUX31 ,8H DAUX32 ,8H DAUX33 ,8H FSMSOL ,         ELTIME
     *   8H PLELP  ,8H BELTRT ,8H SEGSEG ,8H AIRBAG ,8H RSTART ,         ELTIME
     *   8H SETUP2 ,8H IMPULS ,8H IMPLS2 ,8H AIRBG3 ,8H DAUX55 ,         ELTIME
     *   8H EJOINT ,8H SPDAMP ,8H DAUX44 ,8H FLXSEG ,8H EQUILB ,         ELTIME
     *   8H POSTPR ,8H WINDY  ,8H HBELT  ,8H HPTURB ,8H        /         ELTIME
      IF (N.GT.1) GO TO 20                                               ELTIME
      IF (L.GT.1) GO TO 40                                               ELTIME
C                                                                        ELTIME
C     INITIAL CALL AT BEGINNING OF MAIN PROGRAM.                        ELTIME
C                                                                        ELTIME
      MTIN(1) = LTIME(0)                                                 ELTIME
      DO 11 I=1,40                                                       ELTIME
      IND(I)  =  0                                                       ELTIME
      NC(I)   =  0                                                       ELTIME
      MTIN(I) = -1                                                       ELTIME
   11 NT(I)   =  0                                                       ELTIME
      NSUB    =  1                                                       ELTIME
      IND(1)  =  1                                                       ELTIME
      NC(1)   =  1                                                       ELTIME
      MTIN(1) =  0                                                       ELTIME
      GO TO 99                                                           ELTIME
C                                                                        ELTIME
C      CALL AT BEGINNING OF NTH SUBROUTINE.                              ELTIME
C                                                                        ELTIME
   20 IF (L.GT.1) GO TO 30                                               ELTIME
      MTIN(N) = LTIME(1)                                                 ELTIME
      IF (NC(N).NE.0) GO TO 21                                           ELTIME
      NSUB = NSUB+1                                                      ELTIME
      IND(NSUB) = N                                                      ELTIME
   21 NC(N) = NC(N)+1                                                    ELTIME
      GO TO 99                                                           ELTIME
```

```
C                                                                         ELTIME
C     CALL AT END OF NTH SUBROUTINE.                                      ELTIME
C                                                                         ELTIME
   30 MTOUT = LTIME(1)                                                    ELTIME
      NDIFF = MTOUT-MTIN(N)                                               ELTIME
      MTIN(N) = -1                                                        ELTIME
      IF (NDIFF.EQ.0) GO TO 32                                            ELTIME
      NT(N) = NT(N) + NDIFF                                               ELTIME
      DO 31 I=1,40                                                        ELTIME
      IF (MTIN(I).NE.-1) MTIN(I) = MTIN(I) + NDIFF                        ELTIME
   31 CONTINUE                                                            ELTIME
   32 GO TO 99                                                            ELTIME
C                                                                         ELTIME
C     SUBSEQUENT CALLS FROM MAIN PROGRAM, PRINT SUMMARY TABLE.            ELTIME
C                                                                         ELTIME
   40 NTSUM = LTIME(1)                                                    ELTIME
      NT(1) = NTSUM - MTIN(1)                                             ELTIME
      TIME = FLOAT(NTSUM)/100.0                                           ELTIME
      WRITE (6,41) TIME,L                                                 PAGE
   41 FORMAT('1 ELAPSED CPU TIME =',F10.2,' SECONDS',85X,'PAGE',I5//      PAGE
     *    '     SUB         CALLS       TIME        %     '//)            ELTIME
      PCSUM = 0.0                                                         ELTIME
      NTSUM = 0                                                           ELTIME
      DO 42 I=1,NSUB                                                      ELTIME
      J = IND(I)                                                          ELTIME
      PC = FLOAT(NT(J))/TIME                                              ELTIME
      PCSUM = PCSUM + PC                                                  ELTIME
      NTSUM = NTSUM + NT(J)                                               ELTIME
   42 WRITE (6,43) SUB(J),NC(J),NT(J),PC                                  ELTIME
   43 FORMAT(A10,2I10,F10.2)                                              ELTIME
      WRITE (6,44) NTSUM,PCSUM                                            ELTIME
   44 FORMAT('0TOTAL',14X,I10,F10.2)                                      ELTIME
   99 RETURN                                                              ELTIME
      END                                                                 ELTIME
```

```
      SUBROUTINE EQUILB (YPR,IYPR)                                      EQUILB
C                                                REV IV    02/01/88MISDOT
C                                                                       EQUILB
C     ADJUSTS INITIAL INPUT POSITION PARAMETERS SUPPLIED ON CARDS G.2   EQUILB
C     AND G.3 SUCH THAT INITIAL NORMAL CONTACT FORCES ARE EQUAL TO      EQUILB
C     EITHER SUPPLIED VALUES OR THOSE COMPUTED BY CONSTRAINT FORCES.    EQUILB
C                                                                       EQUILB
      IMPLICIT REAL*8(A-H,O-Z)                                          EQUILB
      DIMENSION  YPR(3,30) , IYPR(4,30)                                 EQUILB
      COMMON/CONTRL/ TIME,NSEG,NJNT,NPL,NBLT,NBAG,NVEH,NGRND,           EQUILB
     *               NS,NQ,NSD,NFLX,NHRNSS,NWINDF,NJNTF,NPRT(36),NPG    PAGE
      COMMON/SGMNTS/ D(3,3,30),WMEG(3,30),WMEGD(3,30),U1(3,30),U2(3,30),EQUILB
     *               SEGLP(3,30),SEGLV(3,30),SEGLA(3,30),NSYM(30)       EQUILB
      COMMON/DESCRP/ PHI(3,30),W(30),RW(30),SR(4,60),HA(3,60),HB(3,60), SLIP
     *               RPHI(3,30),HT(3,3,60),SPRING(5,90),VISC(7,90),     EQUILB
     *               JNT(30),IPIN(30),ISING(30),IGLOB(30),JOINTF(30)    EQUILB
      COMMON/CMATRX/ V1(3,30),V2(3,30),V3(3,12),B12(3,3,60),A22(3,3,60),EQUILB
     *               F(3,30),TQ(3,30),WJ(30),A11(3,3,30)                SLIP
      COMMON/CSTRNT/ A13(3,3,24),A23(3,3,24),B31(3,3,24),B32(3,3,24),   EQUILB
     *               HHT(3,3,12),RK1(3,12),RK2(3,12),QQ(3,12),TQQ(3,12),EQUILB
     *               RQQ(3,12),HQQ(3,12),SQQ(12),CFQQ(12),              EQUILB
     *               KQ1(12),KQ2(12),KQTYPE(12)                         EQUILB
      COMMON/TABLES/MXNTI,MXNTB,MXTB1,MXTB2,NTI(50),NTAB(1250),TAB(4500)DIMENB
      COMMON/JBARTZ/ MNPL(   30),MNBLT(   8),MNSEG(   30),MNBAG(   6),  EQUILB
     *               MPL(3,5,30),MBLT(3,5,8),MSEG(3,5,30),MBAG(3,10,6), EQUILB
     *               NTPL( 5,30),NTBLT( 5,8),NTSEG( 5,30)               EQUILB
      COMMON/CNTSRF/ PL(24,30),BELT(20,8),TPTS(6,8),BD(24,40)           EDGE
      COMMON/CNSNTS/ PI,RADIAN,G,THIRD,EPS(24),                         EQUILB
     *               UNITL,UNITM,UNITT,GRAVTY(3),TWOPI                  TWOPI
      COMMON/TITLES/ DATE(3),COMENT(40),VPSTTL(20),BDYTTL(5),           EQUILB
     *               BLTTTL(5,8),PLTTL(5,30),BAGTTL(5,6),SEG(30),       EQUILB
     *               JOINT(30),CGS(30),JS(30)                           EQUILB
      COMMON/FORCES/PSF(7,70),BSF(4,20),SSF(10,40),BAGSF(3,20),         EDGE
     *               PRJNT(7,30),NPANEL(5),NPSF,NBSF,NSSF,NBGSF         EDGE
      REAL DATE,COMENT,VPSTTL,BDYTTL,BLTTTL,PLTTL,BAGTTL,SEG,JOINT      EQUILB
      LOGICAL*1 CGS,JS                                                  EQUILB
      COMMON/TEMPVS/ DMNT(3,3),XMN(3),XMM(3),TM(3),RM(3)                EDGE
      DIMENSION  TEMP(3),T(5),FX(10),FX1(10)                            EDGE
      DIMENSION  X(10),GX(10),DX(10),DXP(10),DPN(5,10)                  EQUILB
      DIMENSION  JPL(10),JSG(10),JX(10),M1(10),M2(10),M3(10),MT(10)     EQUILB
      DIMENSION  NTV(10),NI1(10),NSG(10),NAV(10),KSG(5,10)              EQUILB
      DIMENSION  ISG(5),IPL(5),LTYPE(5),INDGX(5),NTNQ(5)                EQUILB
      DIMENSION  SX(10),SGX(10),XDEV(10),WORD(2)                        EQUILB
      DATA       BLANK/'      '/ , WORD/' SEGLP' , '   YPR'/            EQUILB
      CALL ELTIME (1,35)                                                EQUILB
C                                                                       EQUILB
C     INPUT CARDS G.4, G.5.A-G.5.N, AND G.6.A-G.6.M                     EQUILB
C                                                                       EQUILB
      READ   (5,60)  NVAR,NCON                                          EQUILB
      WRITE  (6,51)  NVAR,NCON,NPG                                      PAGE
```

```
      NPG=NPG+1                                                          PAGE
   51 FORMAT('1',5X,'NVAR =',I3,3X,'NCON =',I3,96X,                      PAGE
     *        'PAGE',I5/120X,'CARD  G.4'/)                               PAGE
      ICARD = 4                                                          EQUILB
      JCARD = 0                                                          EQUILB
      IF  (NVAR.LT.1 .OR. NVAR.GT.10)  GO TO 65                          EQUILB
      IF  (NCON.LT.0 .OR. NCON.GT.5 )  GO TO 65                          EQUILB
      WRITE  (6,52)                                                      EQUILB
   52 FORMAT('0',4X,'J',4X,'NTV',3X,'NI1',3X,'NSG',8X,'GX',12X,'XDEV',  EQUILB
     *7X,'JPL',3X,'JSG',3X,'NAV',3X,'KSG(I,J),I=1,NAV',29X,'CARDS G.5'/)EQUILB
      ICARD = 5                                                          EQUILB
      DO  58  J=1,NVAR                                                   EQUILB
      JCARD = J                                                          EQUILB
      READ   (5,53)     NTV(J),NI1(J),NSG(J),GX(J),XDEV(J),             EQUILB
     *                 JPL(J),JSG(J),IAV,(KSG(I,J),I=1,IAV)              EQUILB
   53 FORMAT(3I4,2F8.0,8I4)                                              EQUILB
      NAV(J) = IAV                                                       EQUILB
      WRITE  (6,54)  J,NTV(J),NI1(J),NSG(J),GX(J),XDEV(J),              EQUILB
     *                 JPL(J),JSG(J),IAV,(KSG(I,J),I=1,IAV)              EQUILB
   54 FORMAT(4I6,2F15.6,8I6)                                             EQUILB
      IF  (NTV(J).LT.1 .OR. NTV(J).GT.2    )  GO TO 65                   EQUILB
      IF  (NI1(J).LT.1 .OR. NI1(J).GT.3    )  GO TO 65                   EQUILB
      IF  (NSG(J).LT.1 .OR. NSG(J).GT.NSEG)  GO TO 65                    EQUILB
      IF  (NAV(J).LT.0 .OR. NAV(J).GT.5    )  GO TO 65                   EQUILB
      IF  (JPL(J).LT.1 .OR. JPL(J).GT.NPL )  GO TO 65                    EQUILB
      IF  (JSG(J).LT.1 .OR. JSG(J).GT.NSEG)  GO TO 65                    EQUILB
      K = JPL(J)                                                         EQUILB
      NNPL = MNPL(K)                                                     EQUILB
      IF  (NNPL.LT.1 .OR. NNPL.GT.5)  GO TO 65                           EQUILB
      DO  55  I=1,NNPL                                                   EQUILB
      IF  (JSG(J).NE.MPL(2,I,K))  GO TO 55                               EQUILB
      JSG(J) = I                                                         EQUILB
      GO TO 56                                                           EQUILB
   55 CONTINUE                                                           EQUILB
      GO TO 65                                                           EQUILB
   56 IF  (NAV(J).LE.0)  GO TO 58                                        EQUILB
      DO  57  I=1,IAV                                                    EQUILB
      IF  (KSG(I,J).LT.1 .OR. KSG(I,J).GT.NSEG)  GO TO 65                EQUILB
   57 CONTINUE                                                           EQUILB
   58 CONTINUE                                                           EQUILB
      IF  (NCON.LE.0)  GO TO 17                                          EQUILB
      WRITE  (6,59)                                                      EQUILB
   59 FORMAT('0',4X,'I',4X,'IPL',3X,'ISG',2X,'LTYPE',2X,'INDGX',         EQUILB
     *       87X,'CARDS G.6'/)                                           EQUILB
      ICARD = 6                                                          EQUILB
      DO  64  I=1,NCON                                                   EQUILB
      JCARD = I                                                          EQUILB
      READ   (5,60)     IPL(I),ISG(I),LTYPE(I),INDGX(I)                  EQUILB
      WRITE  (6,61)  I,IPL(I),ISG(I),LTYPE(I),INDGX(I)                   EQUILB
   60 FORMAT(4I4)                                                        EQUILB
```

```
   61 FORMAT(5I6)                                                      EQUILB
      IF  (  IPL(I).LT.1 .OR.   IPL(I).GT.NPL )  GO TO 65              EQUILB
      IF  (  ISG(I).LT.1 .OR.   ISG(I).GT.NSEG)  GO TO 65              EQUILB
      IF  (LTYPE(I).LT.3 .OR. LTYPE(I).GT.4   )  GO TO 65              EQUILB
      IF  (INDGX(I).LT.0 .OR. INDGX(I).GT.NVAR)  GO TO 65              EQUILB
      J = IPL(I)                                                       EQUILB
      NNPL = MNPL(J)                                                   EQUILB
      IF  (NNPL.LT.1 .OR. NNPL.GT.5)  GO TO 65                         EQUILB
      DO  62  K=1,NNPL                                                 EQUILB
      IF  (ISG(I).NE.MPL(2,K,J))  GO TO 62                             EQUILB
      ISG(I) = K                                                       EQUILB
      GO TO 63                                                         EQUILB
   62 CONTINUE                                                         EQUILB
      GO TO 65                                                         EQUILB
   63 IF  (INDGX(I).LE.0)  GO TO 64                                    EQUILB
      K = INDGX(I)                                                     EQUILB
      IF  (IPL(I).NE.JPL(K) .OR. ISG(I).NE.JSG(K))  GO TO 65           EQUILB
   64 CONTINUE                                                         EQUILB
      GO TO 17                                                         EQUILB
C                                                                      EQUILB
C     INPUT ERROR - PRINT MESSAGE AND TERMINATE PROGRAM.               EQUILB
C                                                                      EQUILB
   65 WRITE  (6,66)  ICARD,JCARD                                       EQUILB
   66 FORMAT('0  INPUT ERROR ON CARD G.',I2,'.',I2,                    EQUILB
     *        '.  PROGRAM TERMINATED.')                                EQUILB
      STOP 26                                                          EQUILB
C                                                                      EQUILB
C     DATA INITIALIZATION.                                             EQUILB
C                                                                      EQUILB
   17 NQORG = NQ                                                       EQUILB
      DO  19  K=1,NVAR                                                 EQUILB
      J = JPL(K)                                                       EQUILB
      I = JSG(K)                                                       EQUILB
      M1(K) = MPL(1,I,J)                                               EQUILB
      M2(K) = MPL(2,I,J)                                               EQUILB
      M3(K) = MPL(3,I,J)                                               EQUILB
      MT(K) = NTPL (I,J)                                               EQUILB
      JX(K) = 1                                                        EQUILB
      DXP(K) = 0.0                                                     EQUILB
      I1 = NI1(K)                                                      EQUILB
      I2 = NSG(K)                                                      EQUILB
      IF  (NTV(K).EQ.1)  X(K) = SEGLP(I1,I2)                           EQUILB
      IF  (NTV(K).EQ.2)  X(K) =    YPR(I1,I2)                          EQUILB
      SX (K) =    X(K)                                                 EQUILB
      SGX(K) =  GX(K)                                                  EQUILB
      IF  (NAV(K).LE.0)  GO TO 19                                      EQUILB
      IAV = NAV(K)                                                     EQUILB
      DO  18  L=1,IAV                                                  EQUILB
      J2 = KSG(L,K)                                                    EQUILB
      IF  (NTV(K).EQ.1)  DPN(L,K) = SEGLP(I1,I2) - SEGLP(I1,J2)        EQUILB
```

```
   18 IF  (NTV(K).EQ.2)  DPN(L,K) =   YPR(I1,I2) -   YPR(I1,J2)           EQUILB
   19 CONTINUE                                                            EQUILB
      IF  (NPRT(27).EQ.0)  GO TO 20                                       EQUILB
C                                                                         EQUILB
C     LET'S SEE WHAT USER INPUT LOOKS LIKE.                               EQUILB
C                                                                         EQUILB
      CALL OUTPUT(0)                                                      EQUILB
      CALL DAUX(0)                                                        EQUILB
      CALL PRINT(6H USER )                                                EQUILB
      CALL OUTPUT(1)                                                      EQUILB
C                                                                         EQUILB
C     START FDF FORCE -> CONSTRAINT FORCE ITERATION                       EQUILB
C                                                                         EQUILB
   20 PENDOT = 0.0                                                        EQUILB
      DO  50  JITTER=1,10                                                 EQUILB
C                                                                         EQUILB
C     ITERATE INPUT (X) SUCH THAT F(X) = G(X)                             EQUILB
C                                                                         EQUILB
      MVAR = 2                                                            EQUILB
      IF  (NVAR.EQ.1)  MVAR = 1                                           EQUILB
      DO  32  M=1,2                                                       EQUILB
      DO  32  I=MVAR,NVAR                                                 EQUILB
      DO  32  J=1,I                                                       EQUILB
      NITER = 10                                                          EQUILB
      IF  (DXP(J).EQ.0.0)  NITER = 50                                     EQUILB
      DX(J) = 0.25                                                        EQUILB
      N1 = M1(J)                                                          EQUILB
      N2 = M2(J)                                                          EQUILB
      N3 = M3(J)                                                          EQUILB
      NP = JPL(J)                                                         EQUILB
      NT = MT(J)                                                          EQUILB
      I1 = NI1(J)                                                         EQUILB
      I2 = NSG(J)                                                         EQUILB
      IAV = NAV(J)                                                        EQUILB
      IF  (NTV(J).NE.2)  GO TO 15                                         EQUILB
      CALL DRCIJK (D,YPR,IYPR,HT,I2)                                      EQUILB
      IF  (NAV(J).LE.0)  GO TO 15                                         EQUILB
      DO  14  K=1,IAV                                                     EQUILB
      J2 = KSG(K,J)                                                       EQUILB
   14 CALL DRCIJK (D,YPR,IYPR,HT,J2)                                      EQUILB
   15 DO  29  ITER=1,NITER                                                EQUILB
      CALL CHAIN(0)                                                       JDRIFT
      PEN1 = PEN                                                          EQUILB
      NPSF = 1                                                            EDGE
      CALL PLELP(N2,N3,N1,NP,NT)                                          MISDOT
      PEN = PSF(1,1)                                                      EDGE
      FX1(J) = FX(J)                                                      EQUILB
      FXJ = 0.0                                                           EQUILB
      IF (PEN.GT.0.0) FXJ = PSF(2,1)                                      EDGE
      IF  (PEN.GT.0.0)  CALL FRCDFL (PEN,PENDOT,NT,1,FXJ,ELOSS)           EQUILB
```

```
      FX(J) = FXJ                                                       EQUILB
      IF  (JX(J)-2)  23,21,25                                           EQUILB
   21 IF  (FX(J)*FX1(J).GT.0.0)  GO TO 22                               EQUILB
      IF  (FX1(J).EQ.0.0)  JX(J) = 1                                    EQUILB
      FX(J) = FX1(J)                                                    EQUILB
      PEN = PEN1                                                        EQUILB
      DX(J) = 0.5*DX(J)                                                 EQUILB
      X(J)  = X(J) - DX(J)                                              EQUILB
      GO TO 27                                                          EQUILB
   22 F2 = FX(J) - GX(J)                                                EQUILB
      F1 = FX1(J) - GX(J)                                               EQUILB
      IF  (F1*F2.LE.0.0)  GO TO 24                                      EQUILB
      IF  (DABS(F2).LT.DABS(F1)) GO TO 23                               EQUILB
   26 FX(J) = FX1(J)                                                    EQUILB
      DX(J) = -DX(J)                                                    EQUILB
      PEN = PEN1                                                        EQUILB
      X(J) = X(J) + 2.0*DX(J)                                           EQUILB
      GO TO 27                                                          EQUILB
   23 JX(J) = 1                                                         EQUILB
      IF  (PEN.GT.0.0)  JX(J) = 2                                       EQUILB
      IF  (ITER.GT.1 .AND. PEN.LT.0.0 .AND. PEN.LT.PEN1)  GO TO 26      EQUILB
      X(J) = X(J) + DX(J)                                               EQUILB
      GO TO 27                                                          EQUILB
   24 DXP(J) = DX(J)/(FX(J)-FX1(J))                                     EQUILB
      JX(J) = 3                                                         EQUILB
   25 IF  (DABS(FX(J)-GX(J)).LT.EPS(6))  GO TO 30                       EQUILB
      IF  (PEN.LT.0.0)  CALL FRCDFL (-PEN,PENDOT,NT,1,FXJ,ELOSS)        EQUILB
      IF  (PEN.LT.0.0)  FX(J) = -FXJ                                    EQUILB
      X(J) = X(J) - DXP(J)*(FX(J)-GX(J))                                EQUILB
   27 IF  (XDEV(J).LE.0.0)  GO TO 42                                    EQUILB
      IF  (DABS(X(J)-SX(J)).LE.XDEV(J))  GO TO 42                       EQUILB
      WRITE  (6,41)  J,X(J),SX(J),XDEV(J)                               EQUILB
   41 FORMAT('0 PROGRAM IS BEING TERMINATED IN SUBROUTINE EQUILB.'//    EQUILB
     *       '  ITERATION FOR VARIABLE NO.',I3,' IS NOT CONVERGING.'//  EQUILB
     *       '  VALUE OF X IS OUT OF RANGE. VALUES OF X,SX,XDEV ARE'//  EQUILB
     *        3G20.8)                                                   EQUILB
      STOP 27                                                           EQUILB
   42 IF  (NTV(J).EQ.1)  SEGLP(I1,I2) = X(J)                            EQUILB
      IF  (NTV(J).EQ.2)    YPR(I1,I2) = X(J)                            EQUILB
      IF  (NTV(J).EQ.2)  CALL DRCIJK (D,YPR,IYPR,HT,I2)                 EQUILB
      IF  (NAV(J).LE.0)  GO TO 29                                       EQUILB
      DO  28  K=1,IAV                                                   EQUILB
      J2 = KSG(K,J)                                                     EQUILB
      IF  (NTV(J).EQ.1)  SEGLP(I1,J2) = X(J) - DPN(K,J)                 EQUILB
      IF  (NTV(J).EQ.2)    YPR(I1,J2) = X(J) - DPN(K,J)                 EQUILB
   28 IF  (NTV(J).EQ.2)  CALL DRCIJK (D,YPR,IYPR,HT,J2)                 EQUILB
   29 CONTINUE                                                          EQUILB
   30 IF  (NPRT(27).NE.0)  WRITE  (6,31)  M,I,J,ITER,X(J),FX(J)         EQUILB
   31 FORMAT(4I3,4X,2F12.6)                                             EQUILB
   32 CONTINUE                                                          EQUILB
```

```
      IF  (NQ.LE.0)  GO TO 40                                          EQUILB
C                                                                       EQUILB
C     COMPUTE VEHICLE COORDINATES FOR FIXED POINT CONSTRAINTS.          EQUILB
C                                                                       EQUILB
      DO  35  K=1,NQ                                                    EQUILB
      IF  (KQTYPE(K).NE.1)  GO TO 35                                    EQUILB
      IF  (KQ2(K).NE.NVEH)  GO TO 35                                    EQUILB
      L = KQ1(K)                                                        EQUILB
      CALL DOT31(D(1,1,L),RK1(1,K),T)                                   EQUILB
      DO  34  I=1,3                                                     EQUILB
   34 T(I) = T(I) + SEGLP(I,L) - SEGLP(I,NVEH)                          EQUILB
      CALL MAT31(D(1,1,NVEH),T,RK2(1,K))                                EQUILB
   35 CONTINUE                                                          EQUILB
   40 IF  (NPRT(27).EQ.0)  GO TO 36                                     EQUILB
C                                                                       EQUILB
C     SOLVE SYSTEM EQUATIONS WITH CONSTRAINTS OFF.                      EQUILB
C                                                                       EQUILB
      CALL OUTPUT(0)                                                    EQUILB
      CALL DAUX(0)                                                      EQUILB
      CALL PRINT(6HEQUIL2)                                              EQUILB
      CALL OUTPUT(1)                                                    EQUILB
C                                                                       EQUILB
C     SET UP CONSTRAINTS TO PRODUCE ZERO ACCELERATIONS.                 EQUILB
C                                                                       EQUILB
   36 NQ = NQORG                                                        EQUILB
      IF  (NCON.LE.0)  GO TO 81                                         EQUILB
      DO  37  I=1,NCON                                                  EQUILB
      NQ = NQ+1                                                         EQUILB
      J = IPL(I)                                                        EQUILB
      K = ISG(I)                                                        EQUILB
      NT = NTPL(K,J)                                                    EQUILB
      NTNQ(I) = NTAB(NT+1)                                              EQUILB
      NTAB(NT+1) = -NQ                                                  EQUILB
      KQ1(NQ) = MPL(2,K,J)                                              EQUILB
      KQ2(NQ) = MPL(1,K,J)                                              EQUILB
   37 KQTYPE(NQ) = LTYPE(I)                                             EQUILB
C                                                                       EQUILB
C     SOLVE SYSTEM EQUATIONS WITH CONSTRAINTS ON.                       EQUILB
C                                                                       EQUILB
      CALL OUTPUT(0)                                                    EQUILB
      CALL DAUX(0)                                                      EQUILB
      IF  (NPRT(27).NE.0.AND.JITTER.EQ.1)  CALL PRINT(6HEQUIL1)         EQUILB
C                                                                       EQUILB
C     FETCH CONSTRAINTS FORCES NORMAL TO PLANE SURFACES.                EQUILB
C     STORE FRICTION FORCE AND TURN OFF CONSTRAINTS.                    EQUILB
C                                                                       EQUILB
      CONV = 1.0                                                        EQUILB
      DO  39  I=1,NCON                                                  EQUILB
      MQ = NQORG+I                                                      EQUILB
      J = IPL(I)                                                        EQUILB
```

```
      K = ISG(I)                                                        EQUILB
      NT = NTPL(K,J)                                                    EQUILB
      NTAB(NT+1) = NTNQ(I)                                              EQUILB
      M = MPL(2,K,J)                                                    EQUILB
      N = MPL(1,K,J)                                                    EQUILB
      CALL DOT31(D(1,1,N),PL(1,J),TEMP)                                 EQUILB
      T(I) = TEMP(1)*QQ(1,MQ) + TEMP(2)*QQ(2,MQ)+ TEMP(3)*QQ(3,MQ)      EQUILB
      I1 = INDGX(I)                                                     EQUILB
      IF  (I1.GT.0 .AND. DABS(GX(I1)+T(I)).GT.EPS(2))  CONV = 0.0       EQUILB
      IF  (I1.GT.0)  GX(I1) = 0.5*(GX(I1)-T(I))                         EQUILB
      DO  38  L=1,3                                                     EQUILB
   38 TEMP(L) = QQ(L,MQ) - T(I)*TEMP(L)                                 EQUILB
      LT = NTAB(NT)                                                     EQUILB
   39 CALL MAT31(D(1,1,M),TEMP,TAB(LT+19))                              EQUILB
      NQ = NQORG                                                        EQUILB
      IF  (CONV.EQ.1.0)  GO TO 81                                       EQUILB
   50 CONTINUE                                                          EQUILB
C                                                                       EQUILB
C     PRINT INPUT AND CHANGES MADE.                                     EQUILB
C                                                                       EQUILB
   81 IF  (NJNT.LE.0)  GO TO 86                                         EQUILB
      CALL OUTPUT(0)                                                    EQUILB
      CALL DAUX(0)                                                      EQUILB
      IPRINT = 0                                                        EQUILB
      DO  84  J=1,NJNT                                                  EQUILB
      IF  (IPIN(J).GE.0)  GO TO 84                                      EQUILB
      IF  (VISC(4,3*J-2).GT.0.0)  GO TO 84                              EQUILB
      IF  (IPIN(J).EQ.-1)  T1 = DABS(XDY(HB(1,2*J),D(1,1,J+1),TQ(1,J))) EQUILB
      IF  (IPIN(J).LE.-2)  T1 = DSQRT(TQ(1,J)**2+TQ(2,J)**2+TQ(3,J)**2) EQUILB
      VISC(4,3*J-2) = 1.5*T1                                            EQUILB
      IF  (IPRINT.EQ.0)  WRITE  (6,82)                                  EQUILB
   82 FORMAT('0  THE FOLLOWING VALUES FOR THE MAX TORQUE FOR A LOCKED JOEQUILB
     *INT ON CARDS B.5 HAVE BEEN SET UP BY SUBROUTINE EQUILB:'//         EQUILB
     *        '        J SYM     IPIN     T1=VISC(4)' /)                 EQUILB
      IPRINT = 1                                                        EQUILB
      WRITE  (6,83)  J,JOINT(J),IPIN(J),VISC(4,3*J-2)                   EQUILB
   83 FORMAT(I6,1X,A4,I6,F15.6)                                         EQUILB
   84 CONTINUE                                                          EQUILB
   86 IF  (NQ.LE.0)  GO TO 91                                           EQUILB
      IPRINT = 0                                                        EQUILB
      DO  89  K=1,NQ                                                    EQUILB
      IF  (KQTYPE(K).NE.1)  GO TO 89                                    EQUILB
      IF  (KQ2(K).NE.NVEH)  GO TO 89                                    EQUILB
      IF  (IPRINT.EQ.0)  WRITE  (6,87)                                  EQUILB
   87 FORMAT('0  THE FOLLOWING VALUES FOR RK2 ON CARDS D.6 FOR FIXED POIEQUILB
     *NT CONSTRAINTS HAVE BEEN CHANGED BY SUBROUTINE EQUILB:'//         EQUILB
     *        5X,'K',3X,'KQTYPE',4X,'KQ1',5X,'KQ2',8X,'RK2(X)',         EQUILB
     *        9X,'RK2(Y)',9X,'RK2(Z)'/)                                 EQUILB
      IPRINT = 1                                                        EQUILB
      WRITE  (6,88)  K,KQTYPE(K),KQ1(K),KQ2(K),(RK2(I,K),I=1,3)         EQUILB
```

```
88  FORMAT(I6,3I8,3F15.6)                                           EQUILB
89  CONTINUE                                                        EQUILB
91  WRITE  (6,92)                                                   EQUILB
92  FORMAT('0    THE FOLLOWING VARIABLES ON CARDS G.2 AND G.3 ',    EQUILB
   *       'HAVE BEEN CHANGED BY SUBROUTINE EQUILB:'//)             EQUILB
    DO  95  J=1,NVAR                                                EQUILB
    I0 = NTV(J)                                                     EQUILB
    I1 = NI1(J)                                                     EQUILB
    I2 = NSG(J)                                                     EQUILB
    WRITE  (6,93)  WORD(I0),I1,I2,SX(J),X(J),BLANK,J,SGX(J),GX(J)   EQUILB
93  FORMAT(4X,A6,'(',I2,',',I2,') FROM',F12.6,' TO',F12.6,          EQUILB
   *          A4,'AND GX(',I2,') FROM',F12.6,' TO',F12.6)           EQUILB
    IF  (NAV(J).LE.0)  GO TO 95                                     EQUILB
    IAV = NAV(J)                                                    EQUILB
    DO  94  I=1,IAV                                                 EQUILB
    J2 = KSG(I,J)                                                   EQUILB
    ZSX = SX(J) - DPN(I,J)                                          EQUILB
    ZXX =  X(J) - DPN(I,J)                                          EQUILB
94  WRITE  (6,93)  WORD(I0),I1,J2,ZSX,ZXX                           EQUILB
95  CONTINUE                                                        EQUILB
    CALL ELTIME (2,35)                                              EQUILB
    RETURN                                                          EQUILB
    END                                                             EQUILB
```

```
      SUBROUTINE EULRAD(D,A,IC)                                         EULRAD
C                                                     REV IV    07/23/86TWOPI
C     COMPUTES EULER ANGLES PRECESSION, NUTATION, AND SPIN IN RADIANS   EULRAD
C     AND PLACES THEM INTO THE A ARRAY FOR GIVEN DIRECTION COSINE MATRIXEULRAD
C                                                                       EULRAD
C       ASSUMES  D = D(S)D(N)D(P) , WHERE                               EULRAD
C                                                                       EULRAD
C           CS SS 0          1  0  0            CP SP 0                 EULRAD
C     D(S)=-SS CS 0   , D(N)= 0  CN SN  , D(P)=-SP CP 0                 EULRAD
C           0  0  1          0 -SN CN            0  0  1                EULRAD
C                                                                       EULRAD
C     AND P=A(1),  N=A(2), S=A(3)                                       EULRAD
C                                                                       EULRAD
C     ROUTINE WILL ALWAYS WORK IN THE MEMORY MODE, I.E., WILL PRODUCE A EULRAD
C     NEW SET OF A'S THAT DIFFER THE LEAST FROM THE INPUTTED A ARRAY.   EULRAD
C     TO USE IN NON-MEMORY MODE,SET ALL A'S TO ZERO, CALL WITH IC = 8.  EULRAD
C                                                                       EULRAD
C     NEW N IS ALWAYS COMPUTED.                                         EULRAD
C     IF N OR PI-N < 10**-6, IC IS USED TO RESOLVE AMBIGUITES ON P & S, EULRAD
C     EXCEPT FOR IC = 2 OR 8 WHERE THEY ARE NOT CHANGED.                EULRAD
C                                                                       EULRAD
      IMPLICIT REAL*8(A-H,O-Z)                                          EULRAD
      DIMENSION A(3),D(3,3),T(6)                                        EULRAD
      COMMON/CNSNTS/ PI,RADIAN,G,THIRD,EPS(24),                         EULRAD
     *               UNITL,UNITM,UNITT,GRAVTY(3),TWOPI                  TWOPI
      IF  (D(3,3).GT. 1.0)  D(3,3) =  1.0                               EULRAD
      IF  (D(3,3).LT.-1.0)  D(3,3) = -1.0                               EULRAD
      B = DACOS(D(3,3))                                                 EULRAD
      T(2) =  B-A(2)                                                    EULRAD
      T(5) = -B-A(2)                                                    EULRAD
      Z = 0.0                                                           EULRAD
      IF  (   B.LT.EPS(6)) Z =  1.0                                     EULRAD
      IF  (PI-B.LT.EPS(6)) Z = -1.0                                     EULRAD
      IF  (Z.NE.0.0)  GO TO 11                                          EULRAD
      T(1) = DATAN2(D(3,1),-D(3,2)) - A(1)                              EULRAD
      T(4) = T(1) + PI                                                  EULRAD
      T(3) = DATAN2(D(1,3), D(2,3)) - A(3)                              EULRAD
      T(6) = T(3) + PI                                                  EULRAD
      GO TO 26                                                          EULRAD
   11 T(1) = DATAN2(D(1,2)-Z*D(2,1) , D(1,1)+Z*D(2,2)) - A(1) - Z*A(3)  EULRAD
      T(3) = T(1)                                                       EULRAD
      GO TO (21,22,23,23,22,21,22,22) , IC                              EULRAD
C                                                                       EULRAD
C     SET T(1) = 0 EXCEPT FOR IC=3,4                                    EULRAD
C     SET T(3) = 0 EXCEPT FOR IC=1,6                                    EULRAD
C                                                                       EULRAD
   21 T(1) = 0.0                                                        EULRAD
      GO TO 25                                                          EULRAD
   22 T(1) = 0.0                                                        EULRAD
   23 T(3) = 0.0                                                        EULRAD
```

```
25    T(4) = T(1)                                                     EULRAD
      T(6) = T(3)                                                     EULRAD
26    TMAX = 0.0                                                      EULRAD
      J    = 3                                                        EULRAD
      DO  30  I=1,6                                                   EULRAD
      T(I) = DMOD(T(I),TWOPI)                                         EULRAD
      IF  (DABS(T(I)).GT.PI  )  T(I) = T(I) - DSIGN(TWOPI,T(I))       EULRAD
      IF  (DABS(T(I)).LT.TMAX)  GO TO 30                              EULRAD
      TMAX = DABS(T(I))                                               EULRAD
      IF  (I.GT.3)  J = 0                                             EULRAD
30    CONTINUE                                                        EULRAD
      IF  (Z.LT.0.0)  T(J+3) = -T(J+3)                                EULRAD
      DO  40  I=1,3                                                   EULRAD
      IJ = I+J                                                        EULRAD
40    A(I) = A(I) + T(IJ)                                             EULRAD
      RETURN                                                          EULRAD
      END                                                             EULRAD
```

```
      DOUBLE PRECISION FUNCTION EVALFD (D,N,L)                          EVALFD
C                                                REV IV    07/23/86JTF786
C     EVALUATE FUNCTION THAT IS DEFINED AT LOCATION N OF TAB ARRAY      EVALFD
C     FOR ABSCISSA VALUE D.  EVALUATES DERIVATIVE, FUNCTION OR INTEGRAL EVALFD
C     AS L EQUALS 0, 1, OR 2. TAB ARRAY IS DEFINED AS FOLLOWS:          EVALFD
C        TAB(N)     -  D0 (NO RESTRICTIONS ON D0)                       JTF786
C        TAB(N+1)   -  D1 (F1 DEFINED FOR D0 < D < !D1!)                JTF786
C        TAB(N+2)   -  D2 (F2 DEFINED FOR !D1! < D < !D2!)              JTF786
C        TAB(N+3)   -  (NOT CURRENTLY USED)                             EVALFD
C        TAB(N+4)   -  (NOT CURRENTLY USED)                             EVALFD
C        TAB(N+5)   -  START OF DEFINITION OF 1ST PART OF FUNCTION (F1) EVALFD
C        WHICH IS FOLLOWED BY DEFINITION OF 2ND PART OF FUNCTION (F2),  EVALFD
C                                IF ANY.                                EVALFD
C        2ND PART OF FUNCTION EXISTS IF D2 IS NON-ZERO.                 EVALFD
C        SIGN OF D1 DETERMINES FORM OF DEFINITION FOR 1ST PART OF       EVALFD
C                               THE FUNCTION.                           EVALFD
C                                                                       EVALFD
C             D1 ZERO INDICATES THAT FUNCTION IS CONSTANT D2 FOR ALL D. EVALFD
C                                                                       EVALFD
C             D1 POSITIVE INDICATES THAT TAB(N+5)-TAB(N+10) CONTAINS    EVALFD
C             A0,A1,...A5.   THE COEFFICIENTS OF A 5TH ORDER POLYNOMIAL.EVALFD
C                                                                       EVALFD
C             D1 NEGATIVE INDICATES THAT TAB(N+5) CONTAINS NP (REAL)    EVALFD
C             FOLLOWED BY   D(1), F(1), D(2), F(2) ..., D(NP), F(NP)    EVALFD
C                                                                       EVALFD
C             WARNING- TABULAR FUNCTION MUST BE DEFINED FOR WHOLE RANGE,EVALFD
C             THAT IS, FROM D0 TO D1 INCLUSIVE,OR D1 TO D2 INCLUSIVE.   EVALFD
C                                                                       EVALFD
C                                                                       EVALFD
C        SIMILARLY, THE SIGN OF D2 (IF NON-ZERO) DETERMINES FORM OF     EVALFD
C        DEFINITION OF 2ND PART OF FUNCTION, IF ANY.                    EVALFD
C                                                                       EVALFD
C                                                                       EVALFD
C     IF D < D0    AND D1# 0, DERIVATIVE = 0 OR FUNCTION = F1(D0)       JTF786
C     IF D > !D1!  AND  D2=0, DERIVATIVE = 0 OR FUNCTION = F1(!D1!)     JTF786
C     IF D > !D2!  AND  D2#0, DERIVATIVE = 0 OR FUNCTION = F2(!D2!)     JTF786
C                                                                       EVALFD
C     NOTE: PREVIOUS VERSIONS ASSUMED THAT D0 WAS NON-NEGATIVE AND      JTF786
C           THAT F = 0 FOR D < D0.                                      JTF786
C                                                                       JTF786
      IMPLICIT REAL*8(A-H,O-Z)                                          EVALFD
      COMMON/TABLES/MXNTI,MXNTB,MXTB1,MXTB2,NTI(50),NTAB(1250),TAB(4500)EVALFD
      F = 0.0                                                           EVALFD
      IOUTR = 0                                                         EVALFD
      D0 = TAB(N)                                                       EVALFD
      D1 = TAB(N+1)                                                     EVALFD
      D2 = TAB(N+2)                                                     EVALFD
      IF  (D1.NE.0.0)  GO TO 26                                         EVALFD
      IF  (L-1)  40,24,25                                               JTF786
   24 F = D2                                                            EVALFD
```

```
      GO TO 40                                                          EVALFD
   25 F= (D-D0)*D2                                                      EVALFD
      GO TO 40                                                          EVALFD
C                                                                       EVALFD
C     COMPUTE INDEX OF F1 DEFINITION                                    EVALFD
C                                                                       EVALFD
   26 NP = N+5                                                          EVALFD
      IF (L.EQ.2) GO TO 41                                              EVALFD
C                                                                       EVALFD
C     DERIVATIVES AND FUNCTIONS HERE, INTEGRALS HAVE OTHER LOGIC        EVALFD
C                                                                       EVALFD
      IF (D.GE.D0) GOTO 22                                              JTF786
C                                                                       JTF786
C     D < D0, RETURN F=0 FOR L=0, OR F=F1(D0) FOR L=1.                  JTF786
C                                                                       JTF786
      IF (L.EQ.0) GOTO 40                                               JTF786
      X = D0                                                            JTF786
      IF (D1.GT.0.0) GOTO 37                                            JTF786
      F = TAB(NP+2)                                                     JTF786
      GOTO 40                                                           JTF786
22    IF (D.LT.DABS(D1)) GOTO 31                                        EVALFD
      IF (D2.NE.0.0) GO TO 32                                           EVALFD
C                                                                       EVALFD
C     D .GE.!D1! , D2 = 0                                               EVALFD
C                                                                       EVALFD
      IF (D1.LE.0.0) GO TO 33                                           EVALFD
C                                                                       EVALFD
C     IOUTR.EQ.1 INDICATES D BEYOND RANGE. DERIVATIVE = 0.              EVALFD
C     IOUTR.EQ.0 INDICATES D.LE.!D1!. COMPUTE POLY DERIVATIVE           EVALFD
C                                                                       EVALFD
      IF (D.GT.DABS(D1)) IOUTR = 1                                      EVALFD
      X = D1                                                            EVALFD
      GO TO 37                                                          EVALFD
C                                                                       EVALFD
C     D0 < D < !D1!                                                     EVALFD
C                                                                       EVALFD
   31 IF (D1.LT.0.0) GO TO 35                                           EVALFD
      X = D                                                             EVALFD
      GO TO 37                                                          EVALFD
C                                                                       EVALFD
C     D .GE. !D1!,  D2 NON-ZERO,  USE F2                                EVALFD
C                                                                       EVALFD
   32 MP = 6                                                            EVALFD
C                                                                       EVALFD
C     COMPUTE INDEX OF F2 DEFINITION                                    EVALFD
C                                                                       EVALFD
      IF (D1.LT.0.0) MP = 2.0 * TAB(NP)+1.0                             EVALFD
      NP = NP+MP                                                        EVALFD
      IF (D.LT.DABS(D2)) GO TO 34                                       EVALFD
      IF (D2.LT.0.0) GO TO 33                                           EVALFD
```

```
C                                                                       EVALFD
C     IOUTR.EQ.1 INDICATES D BEYOND RANGE. DERIVATIVE = 0.              EVALFD
C     IOUTR.EQ.0 INDICATES D.LE.!D2!. COMPUTE POLY DERIVATIVE           EVALFD
C                                                                       EVALFD
      IF (D.GT.DABS(D2)) IOUTR = 1                                      EVALFD
C                                                                       EVALFD
C     D .GE. D2 (POSITIVE). EVALUATE F2 FOR D2                          EVALFD
C                                                                       EVALFD
      X = D2                                                            EVALFD
      GO TO 37                                                          EVALFD
C                                                                       EVALFD
C     D EXCEEDS TABULAR DEFINITION, SET F = F(NP)                       EVALFD
C     IF TABLE DEFINITION EXTENDS BEYOND RANGE, USE TABLE VALUES        EVALFD
C                                                                       EVALFD
   33 MB = TAB(NP)                                                      EVALFD
      NB = NP+MB+MB                                                     EVALFD
      IF (D .LE. TAB(NB-1)) GO TO 35                                    EVALFD
      IF (L.EQ.1) F=TAB(NB)                                             EVALFD
      GO TO 40                                                          EVALFD
C                                                                       EVALFD
C     !D1! .LE. D < !D2!                                                EVALFD
C                                                                       EVALFD
   34 IF (D2.LT.0.0) GO TO 35                                           EVALFD
      X = D                                                             EVALFD
      GO TO 37                                                          EVALFD
C                                                                       EVALFD
C     EVALUATE F FROM TABULAR DEFINITION                                EVALFD
C                                                                       EVALFD
   35 MB = TAB(NP)                                                      EVALFD
      K1 = NP+3                                                         EVALFD
      K2 = NP+MB+MB                                                     EVALFD
      DO 36 K=K1,K2,2                                                   EVALFD
      IF (D.GT.TAB(K)) GO TO 36                                         EVALFD
      IF (L-1) 28,27,40                                                 EVALFD
C                                                                       EVALFD
C     EVALUATE DERIVATIVE FROM TABLE                                    EVALFD
C                                                                       EVALFD
  28   F = (TAB(K+1)-TAB(K-1))/(TAB(K)-TAB(K-2))                        EVALFD
      GO TO 40                                                          EVALFD
C                                                                       EVALFD
C     EVALUATE FUNCTION FROM TABLE                                      EVALFD
C                                                                       EVALFD
  27  R2 = TAB(K)-TAB(K-2)                                              EVALFD
      R1 = (D-TAB(K-2))/R2                                              EVALFD
      R2 = (TAB(K)-D )/R2                                               EVALFD
      F = R1*TAB(K+1)+R2*TAB(K-1)                                       EVALFD
      GO TO 40                                                          EVALFD
   36 CONTINUE                                                          EVALFD
      IF (L.EQ.1) F = TAB(K2)                                           EVALFD
      GO TO 40                                                          EVALFD
```

```
   37 IF (IOUTR.EQ.1 .AND. L.EQ.0 ) GO TO 40                            EVALFD
      IF (L-1) 38,39,40                                                  EVALFD
C                                                                        EVALFD
C     EVALUATE DERIVATIVE OF 5TH DEGREE POLYNOMIAL                       EVALFD
C                                                                        EVALFD
   38 F = TAB(NP+1)+X*(2.0*TAB(NP+2)+X*(3.0*TAB(NP+3)+X*(4.0*TAB(NP+4)+ EVALFD
     *        X*5.0*TAB(NP+5))))                                         EVALFD
      GO TO 40                                                           EVALFD
C                                                                        EVALFD
C     EVALUATE 5TH DEGREE POLYNOMIAL                                     EVALFD
C                                                                        EVALFD
   39 F =     TAB(NP) + X*(TAB(NP+1)+X*(TAB(NP+2)                         EVALFD
     *   +X*(TAB(NP+3)+X*(TAB(NP+4)+X*TAB(NP+5)))))                      EVALFD
      GO TO 40                                                           EVALFD
C                                                                        EVALFD
C     L=2: COMPUTE INTEGRAL OF FUNCTION FROM D0 TO D.                    EVALFD
C                                                                        EVALFD
   41 IF (D.EQ.D0) GO TO 40                                              EVALFD
      X0 = D0                                                            EVALFD
      X1 = D1                                                            EVALFD
      DO 50 I=1,2                                                        EVALFD
      IF (X1) 43,49,42                                                   EVALFD
   42 A0 = TAB(NP  )                                                     EVALFD
      A1 = TAB(NP+1)/2.0                                                 EVALFD
      A2 = TAB(NP+2)/3.0                                                 EVALFD
      A3 = TAB(NP+3)/4.0                                                 EVALFD
      A4 = TAB(NP+4)/5.0                                                 EVALFD
      A5 = TAB(NP+5)/6.0                                                 EVALFD
      NP = NP+6                                                          EVALFD
      X = X0                                                             EVALFD
      IF (X.NE.0.0) F=F-X*(A0+X*(A1+X*(A2+X*(A3+X*(A4+X*A5)))))          EVALFD
      X = DMIN1(D,X1)                                                    EVALFD
      IF (X.NE.0.0) F=F+X*(A0+X*(A1+X*(A2+X*(A3+X*(A4+X*A5)))))          EVALFD
      IF(D.LE.X1)  GO TO 40                                              EVALFD
      IF(I.EQ.1.AND.D2.NE.0.0)  GO TO 49                                 EVALFD
C                                                                        EVALFD
C     NOTE -  NP WAS UPDATED NP=NP+6 BEFORE THIS, READY FOR SECOND PASS EVALFD
C                                                                        EVALFD
      F = F + (D-X1)*(TAB(NP-6)+X1*(TAB(NP-5)+X1*(TAB(NP-4)              EVALFD
     *          +X1*(TAB(NP-3)+X1*(TAB(NP-2)+X1*TAB(NP-1))))))           EVALFD
      GO TO 40                                                           EVALFD
   43 MB = TAB(NP)                                                       EVALFD
      K1 = NP+3                                                          EVALFD
      K2 = NP+MB+MB                                                      EVALFD
      NP = K2+1                                                          EVALFD
      DL = DMIN1(D,DABS(X1))                                             EVALFD
      DO 44 K=K1,K2,2                                                    EVALFD
      IF (X0.GE.TAB(K)) GO TO 44                                         EVALFD
      Z1 = DMAX1(X0,TAB(K-2))                                            EVALFD
      Z2 = DMIN1(DL,TAB(K))                                              EVALFD
```

```
      FYX = TAB(K-1)*TAB(K) - TAB(K+1)*TAB(K-2)                          EVALFD
      FY = TAB(K+1) - TAB(K-1)                                           EVALFD
      F = F +(FYX  + 0.5*FY*(Z1+Z2)) *(Z2-Z1)/ (TAB(K)-TAB(K-2))         EVALFD
      IF (Z2.NE.DL) GO TO 44                                             EVALFD
      IF(I.EQ.1.AND.D2.NE.0.0)  GO TO 49                                 EVALFD
      IF(Z2. EQ. D) GO TO 40                                             EVALFD
      F = F +(D-Z2)*(FYX+Z2*FY)/ (TAB(K)-TAB(K-2))                       EVALFD
      GO TO 40                                                           EVALFD
   44 CONTINUE                                                           EVALFD
   49 X0 = DABS(D1)                                                      EVALFD
   50 X1 = D2                                                            EVALFD
   40 EVALFD = F                                                         EVALFD
      RETURN                                                             EVALFD
      END                                                                EVALFD
```

```
      SUBROUTINE FDINIT                                                 FDINIT
C                                                 REV III.2 08/08/84REVIII
C     REPLACES CODE PREVIOUSLY IN SUBROUTINES FINPUT AND HINPUT.        FDINIT
C     FROM FIVE FUNCTION NUMBERS IN NF ARRAY                            FDINIT
C      1. SET UP KTITLE                                                 FDINIT
C      2. SET UP NTAB AND TAB ARRAYS                                    FDINIT
C      3. INCREMENT COUNTERS MXNTB AND MXTB2                            FDINIT
C                                                                       FDINIT
      IMPLICIT REAL*8 (A-H,O-Z)                                         FDINIT
      COMMON/TABLES/MXNTI,MXNTB,MXTB1,MXTB2,NTI(50),NTAB(1250),TAB(4500)DIMENB
      COMMON/TEMPVS/ JTITLE(5,51),NF(5),MS(3),KTITLE(31)                FDINIT
C     NOTE: THIS IS SHARED BY SUBS CINPUT, FINPUT, HINPUT AND FDINIT.   FDINIT
      REAL JTITLE,KTITLE                                                FDINIT
      J1 = MXTB2 + 1                                                    FDINIT
      NT = MXNTB + 1                                                    FDINIT
      NTAB(NT) = J1                                                     FDINIT
      NT = NT+1                                                         FDINIT
      DO 56 L=1,5                                                       FDINIT
      NX = IABS(NF(L))                                                  FDINIT
      NTAB(NT) = 0                                                      FDINIT
      IF (NX.EQ.0) GO TO 56                                             FDINIT
      NTAB(NT) = ISIGN(NTI(NX),NF(L))                                   FDINIT
      DO 51 KK = 1,5                                                    FDINIT
      KJ = 5*L+KK+1                                                     FDINIT
   51 KTITLE(KJ) = JTITLE(KK,NX)                                        FDINIT
      IF (NTI(NX).NE.0) GO TO 56                                        FDINIT
      WRITE(6,54) NX                                                    FDINIT
   54 FORMAT ('0 FUNCTION NO.',I4,' HAS NOT BEEN DEFINED. ',            FDINIT
     *        '  PROGRAM TERMINATED.')                                  FDINIT
      STOP 15                                                           FDINIT
   56 NT = NT+1                                                         FDINIT
C                                                                       FDINIT
C     INITIALIZE TAB ARRAY TO ZERO EXCEPT FOR DMAX, DINER, FDMAX.       FDINIT
C                                                                       FDINIT
      J2 = J1+29                                                        FDINIT
      DO 57 JJ=J1,J2                                                    FDINIT
   57 TAB(JJ) = 0.0                                                     FDINIT
      NX = NTAB(NT-5)                                                   FDINIT
      IF (NX.LE.0) GO TO 58                                             BUTLER1
      TAB(J1+8) = DABS(TAB(NX+1))                                       FDINIT
      IF (TAB(NX+2).NE.0.0) TAB(J1+8) = DABS(TAB(NX+2))                 FDINIT
      DX = TAB(J1+8)                                                    FDINIT
      TAB(J1+10) = EVALFD(DX,NX,1)                                      FDINIT
      NX = NTAB(NT-4)                                                   FDINIT
      IF (NX.LE.0) GO TO 58                                             FDINIT
      TAB(J1+9) = DABS(TAB(NX+1))                                       FDINIT
      IF (TAB(NX+2).NE.0.0) TAB(J1+9) = DABS(TAB(NX+2))                 FDINIT
   58 J1 = J2+1                                                         FDINIT
      MXNTB = NT-1                                                      FDINIT
      MXTB2 = J1-1                                                      FDINIT
```

```
      IF  (MXTB2.GT.4500)  WRITE (6,62) MXTB2                          DIMENB
   62 FORMAT ('0 ERROR IN SUBROUTINE FDINIT, SIZE OF TAB ARRAY =',I8//  FDINIT
     *        ' PROGRAM TERMINATED.')                                   FDINIT
      IF (MXNTB.GT.1250) WRITE (6,63) MXNTB                             DIMENB
   63 FORMAT ('0 ERROR IN SUBROUTINE FDINIT, SIZE OF NTAB ARRAY =',I8// FDINIT
     *        ' PROGRAM TERMINATED.')                                   FDINIT
      IF  (MXTB2.GT.4500.OR.MXNTB.GT.1250) STOP 16                      DIMENB
      RETURN                                                            FDINIT
      END                                                               FDINIT
```

```
      SUBROUTINE FINPUT                                                 FINPUT
C                                                    REV IV    02/01/88MISDOT
C     INPUT CARDS F.1-F.5 SPECIFYING THE ALLOWED CONTACTS OF THE CRASH  FINPUT
C     VICTIM BODY SEGMENTS WITH VEHICLE PANELS, BELTS, AIRBAGS AND OTHERFINPUT
C     BODY SEGMENTS ALONG WITH THE ASSOCIATED FUNCTIONS TO BE USED FOR  FINPUT
C     EACH CONTACT.                                                     FINPUT
C     ALSO SETS UP TABLES TO CONTROL TIME HISTORY INFORMATION FOR       FINPUT
C     EACH FUNCTION FOR EACH ALLOWED CONTACT.                           FINPUT
C                                                                       FINPUT
      IMPLICIT REAL*8(A-H,O-Z)                                          FINPUT
      COMMON/CONTRL/ TIME,NSEG,NJNT,NPL,NBLT,NBAG,NVEH,NGRND,           FINPUT
     *               NS,NQ,NSD,NFLX,NHRNSS,NWINDF,NJNTF,NPRT(36),NPG    PAGE
      COMMON/DESCRP/ PHI(3,30),W(30),RW(30),SR(4,60),HA(3,60),HB(3,60), SLIP
     *               RPHI(3,30),HT(3,3,60),SPRING(5,90),VISC(7,90),     FINPUT
     *               JNT(30),IPIN(30),ISING(30),IGLOB(30),JOINTF(30)    FINPUT
      COMMON/JBARTZ/ MNPL(   30),MNBLT(   8),MNSEG(   30),MNBAG(    6), FINPUT
     *               MPL(3,5,30),MBLT(3,5,8),MSEG(3,5,30),MBAG(3,10,6), FINPUT
     *               NTPL( 5,30),NTBLT( 5,8),NTSEG( 5,30)               FINPUT
      COMMON/TABLES/MXNTI,MXNTB,MXTB1,MXTB2,NTI(50),NTAB(1250),TAB(4500)DIMENB
      COMMON/TITLES/ DATE(3),COMENT(40),VPSTTL(20),BDYTTL(5),           FINPUT
     *               BLTTTL(5,8),PLTTL(5,30),BAGTTL(5,6),SEG(30),       FINPUT
     *               JOINT(30),CGS(30),JS(30)                           FINPUT
      REAL DATE,COMENT,VPSTTL,BDYTTL,BLTTTL,PLTTL,BAGTTL,SEG,JOINT      FINPUT
      LOGICAL*1 CGS,JS                                                  FINPUT
      COMMON/CSTRNT/ A13(3,3,24),A23(3,3,24),B31(3,3,24),B32(3,3,24),   FINPUT
     *               HHT(3,3,12),RK1(3,12),RK2(3,12),QQ(3,12),TQQ(3,12),FINPUT
     *               RQQ(3,12),HQQ(3,12),SQQ(12),CFQQ(12),              FINPUT
     *               KQ1(12),KQ2(12),KQTYPE(12)                         FINPUT
      COMMON/WINDFR/ WTIME(30),QFU(3,5),QFV(3,5),WF(3,30),IWIND(30),    WINDOP
     *               MWSEG(7,30),NFVSEG(6),NFVNT(5),MOWSEG(30,30)       WINDOP
      COMMON/TEMPVS/JTITLE(5,51),NF(5),MS(3),KTITLE(31)                 FINPUT
C                                                                       FINPUT
      REAL  JTITLE,KTITLE,BLANK,SURFCE(2,3)                             FINPUT
      DATA  BLANK/4H    /                                               FINPUT
      DATA  SURFCE/4H  PL,4HANE ,4H  BE,4HLT  ,4H SEG,4HMENT/           FINPUT
C                                                                       FINPUT
      MXNTI = 50                                                        FINPUT
      MXNTB = 0                                                         FINPUT
      MXTB2 = MXTB1                                                     FINPUT
C                                                                       FINPUT
C     INPUT ALLOWED CONTACTS AND FUNCTIONS BY REF. NO.                  FINPUT
C                                                                       FINPUT
      WRITE (6,31) NPG                                                  PAGE
      NPG=NPG+1                                                         PAGE
   31 FORMAT('1 ALLOWED CONTACTS AND ASSOCIATED FUNCTIONS',80X,         PAGE
     *        'PAGE',I5)                                                PAGE
      DO 61 I=1,4                                                       FINPUT
      IJK = 0                                                           FINPUT
      GO TO (32,34,35,36),I                                             FINPUT
   32 IF (NPL.LE.0) GO TO 61                                            FINPUT
```

```
C                                                                        FINPUT
C     INPUT NO. OF SEGMENTS TO CONTACT EACH PLANE.                       FINPUT
C     INPUT CARD F.1.A                                                   FINPUT
C                                                                        FINPUT
      READ  (5,33) (MNPL(J),J=1,NPL)                                     FINPUT
   33 FORMAT(18I4)                                                       FINPUT
      NJJ = NPL                                                          FINPUT
      GO TO 37                                                           FINPUT
   34 IF (NBLT.LE.0) GO TO 61                                            FINPUT
C                                                                        FINPUT
C     INPUT NO. OF SEGMENTS TO CONTACT EACH BELT.                        FINPUT
C     INPUT CARD F.2.A                                                   FINPUT
C                                                                        FINPUT
      READ  (5,33) (MNBLT(J),J=1,NBLT)                                   FINPUT
      NJJ = NBLT                                                         FINPUT
      GO TO 37                                                           FINPUT
   35 IF (NSEG.LE.0) GO TO 61                                            FINPUT
C                                                                        FINPUT
C     INPUT NO. OF SEGMENTS TO CONTACT EACH SEGMENT.                     FINPUT
C     INPUT CARD F.3.A                                                   FINPUT
C                                                                        FINPUT
      READ  (5,33) (MNSEG(J),J=1,NSEG)                                   FINPUT
      NJJ = NSEG                                                         FINPUT
      NSEG1 = NSEG+1                                                     FINPUT
      DO  26  J=NSEG1,NGRND                                              FINPUT
   26 MNSEG(J) = 0                                                       FINPUT
      GO TO 37                                                           FINPUT
   36  IF (NJNT.LE.0) GO TO 61                                           FINPUT
C                                                                        FINPUT
C     INPUT CARD F.4.A                                                   FINPUT
C     SUPPLY IGLOB(J)=1 FOR EACH GLOBALGRAPHIC JOINT J=1,NJNT            FINPUT
C                                                                        FINPUT
       READ (5,33) (IGLOB(J),J=1,NJNT)                                   FINPUT
      NJJ = NJNT                                                         FINPUT
C                                                                        FINPUT
C     START OF LOOP TO READ CONTACTS FOR PLANES (I=1), BELTS (I=2),      FINPUT
C     SEGMENTS (I=3) AND FUNCTIONS FOR GLOBALGRAPHIC JOINTS (I=4).       FINPUT
C                                                                        FINPUT
   37 DO 60 J=1,NJJ                                                      FINPUT
      IF (I.EQ.1) NK = MNPL(J)                                           FINPUT
      IF (I.EQ.2) NK = MNBLT(J)                                          FINPUT
      IF (I.EQ.3) NK = MNSEG(J)                                          FINPUT
      IF (I.EQ.4) NK = IGLOB(J)                                          FINPUT
      IF (NK.LE.0) GO TO 60                                              FINPUT
      DO 59 K=1,NK                                                       FINPUT
      IF (IJK.EQ.0) WRITE (6,38) I                                       FINPUT
   38 FORMAT('0',119X,'CARDS F.',I1)                                     FINPUT
      IF (IJK.EQ.0 .AND. I.NE.4) WRITE (6,39) SURFCE(1,I),SURFCE(2,I)    FINPUT
   39 FORMAT('0',3X,2A4,8X,'SEGMENT',2X,'FORCE DEFLECTION',6X,'INERTIAL FINPUT
     *SPIKE',10X,'R FACTOR',13X,'G FACTOR',10X,'FRICTION COEF. OPT')     EDGE
```

```
      IF (IJK.EQ.0 .AND. I.EQ.4) WRITE (6,40)                           FINPUT
   40 FORMAT('0',5X,'JOINT (GLOBALGRAPHIC)',2X,'TORQUE DEFLECTION',6X,'HFINPUT
     *ERRON FORMULA',10X,'R FACTOR',13X,'G FACTOR',10X,'FRICTION COEF.')FINPUT
      IJK = 1                                                           FINPUT
C                                                                       FINPUT
C     INPUT CONTACT SURFACE NO., SEGMENT NO., AND FUNCTION NOS.         FINPUT
C     INPUT CARD F.(I).(K)                                              FINPUT
C                                                                       FINPUT
      READ  (5,33) NJ,MS,NF,NX                                          EDGE
      WRITE (6,41) NJ,MS,NF,NX                                          EDGE
   41 FORMAT('0',I7,'-',I3,I11,'-',I3,I8,4I21,I12)                      EDGE
      IF (NJ.NE.J) WRITE (6,42)                                         FINPUT
   42 FORMAT(' CONTACT INPUT ERROR. PROGRAM TERMINATED.')               FINPUT
      IF   (NJ.NE.J)  STOP 14                                           FINPUT
      IF (I.NE.2.AND.NF(5).EQ.0) WRITE(6,20)                            MISDOT
20    FORMAT(' FRICTION FUNCTION NUMBER CAN NOT BE ZERO FOR THIS TYPE OFMISDOT
     * CONTACT')                                                        MISDOT
      IF (I.NE.2.AND.NF(5).EQ.0) STOP 105                               MISDOT
      NLT = 1                                                           FINPUT
      DO 43 JJ = 1,31                                                   FINPUT
   43 KTITLE(JJ) = BLANK                                                FINPUT
      GO TO (44,46,48,49),I                                             FINPUT
C                                                                       FINPUT
C     PLACE SEGMENT NO. AND INDEX TO NTAB ARRAY INTO M- AND NT- ARRAYS. FINPUT
C                                                                       FINPUT
   44 MPL(1,K,J) = MS(1)                                                FINPUT
      MPL(2,K,J) = MS(2)                                                FINPUT
      MPL(3,K,J) = MS(3)                                                FINPUT
      NTPL(K,J)  = MXNTB+1                                              FINPUT
      DO 45 JJ = 1,5                                                    FINPUT
   45 KTITLE(JJ) = PLTTL (JJ,J)                                         FINPUT
      GO TO 50                                                          FINPUT
   46 MBLT(1,K,J) = MS(1)                                               FINPUT
      MBLT(2,K,J) = MS(2)                                               FINPUT
      MBLT(3,K,J) = MS(3)                                               FINPUT
      NTBLT(K,J) = MXNTB+1                                              FINPUT
      DO 47 JJ = 1,5                                                    FINPUT
   47 KTITLE(JJ) = BLTTTL (JJ,J)                                        FINPUT
C                                                                       FINPUT
C     SET UP TWO TABLES FOR FULL BELT FRICTION                          FINPUT
C                                                                       FINPUT
      IF (NF(5).NE.0)  NLT = 2                                          FINPUT
      GO TO 50                                                          FINPUT
   48 MSEG(1,K,J) = MS(1)                                               FINPUT
      MSEG(2,K,J) = MS(2)                                               FINPUT
      MSEG(3,K,J) = MS(3)                                               FINPUT
      NTSEG(K,J) = MXNTB+1                                              FINPUT
      KTITLE (3) = SEG(J)                                               FINPUT
      GO TO 50                                                          FINPUT
C                                                                       FINPUT
```

```
C        NOTE:  GLOBALGRAPHIC JOINT WILL SAVE NT IN IGLOB ARRAY               FINPUT
C                                                                             FINPUT
   49 IGLOB(J)    = MXNTB+1                                                   FINPUT
      KTITLE(2) = JOINT(J)                                                    FINPUT
C                                                                             FINPUT
C     SET UP POINTERS TO TAB ARRAY IN NTAB ARRAY.                             FINPUT
C                                                                             FINPUT
   50 NFJ = MS(2)                                                             FINPUT
      IF (NFJ.GT.0) KTITLE(6) = SEG(NFJ)                                      FINPUT
      DO 51 JJ=1,NLT                                                          FINPUT
   51 CALL FDINIT                                                             FINPUT
      WRITE (6,53) KTITLE                                                     FINPUT
   53 FORMAT(1X,5A4,1X,A4,5(1X,5A4))                                          FINPUT
      LT = NTAB(MXNTB-5)                                                      EDGE
      IF (I.EQ.1) TAB(LT+22) = NX                                             EDGE
      IF (NF(1).NE.0) GO TO 59                                                EDGE
C                                                                             FINPUT
C     IF FORCE DEFLECTION FUNCTION NO. IS ZERO,                               FINPUT
C     SET UP FOR ROLLING CONSTRAINT                                           FINPUT
C                                                                             FINPUT
      NQ = NQ+1                                                               FINPUT
      NTAB(MXNTB-4) = -NQ                                                     FINPUT
      KQTYPE(NQ) = -4                                                         FINPUT
      KQ1(NQ) = MS(2)                                                         FINPUT
      KQ2(NQ) = MS(1)                                                         FINPUT
      IF (I.NE.3) GO TO 59                                                    EDGE
      KQ1(NQ) = J                                                             FINPUT
      KQ2(NQ) = MS(2)                                                         FINPUT
   59 CONTINUE                                                                FINPUT
   60 CONTINUE                                                                FINPUT
   61 CONTINUE                                                                FINPUT
C                                                                             FINPUT
C     INPUT CARD F.5 - JOINT FUNCTIONS TO BE USED.                            FINPUT
C                                                                             FINPUT
      IF (NJNT.LE.0) GO TO 81                                                 FINPUT
      IF  (NJNTF.NE.0)  GO TO 76                                              FINPUT
      DO  75  J=1,NJNT                                                        FINPUT
   75 JOINTF(J) = 0                                                           FINPUT
      GO TO 81                                                                FINPUT
   76 READ  (5,33)  (JOINTF(J),J=1,NJNT)                                      FINPUT
      IJK = 0                                                                 FINPUT
      DO 80 J=1,NJNT                                                          FINPUT
      IF (JOINTF(J).EQ.0) GO TO 80                                            FINPUT
      IF (IJK.EQ.0) WRITE (6,77) NPG                                          PAGE
      IF (IJK.EQ.0) NPG=NPG+1                                                 PAGE
   77 FORMAT('1',122X,'PAGE',I5/120X,'CARD F.5'/                              PAGE
     *       ' THE FOLLOWING JOINT RESTORING FORCE FUNCTIONS AS DEFINED FINPUT
     *ON CARDS E.7 WILL BE USED.'//4X,'JOINT',10X,'FUNCTION'//)               FINPUT
      JF = JOINTF(J)                                                          FINPUT
      IJK = 1                                                                 FINPUT
```

```
      WRITE (6,78) J,JOINT(J),JF,(JTITLE(I,JF),I=1,5)                  FINPUT
   78 FORMAT(I6,'-',A4,I10,'-',5A4)                                    FINPUT
      IF (NTI(JF).EQ.0) WRITE (6,42)                                   FINPUT
      IF (NTI(JF).EQ.0) STOP 17                                        FINPUT
   80 CONTINUE                                                         FINPUT
C                                                                      FINPUT
C     INPUT CONTACT SEGMENTS FOR AIRBAG, IF ANY.                       FINPUT
C                                                                      FINPUT
   81 IF (NBAG.LE.0) GO TO 69                                          FINPUT
      IJK = 0                                                          FINPUT
      DO 68 J=1,NBAG                                                   FINPUT
C                                                                      FINPUT
C     INPUT CARD F.6.(J)                                               FINPUT
C                                                                      FINPUT
      READ (5,63) K,NK,(MBAG(2,I,J),MBAG(3,I,J),I=1,NK)                FINPUT
   63 FORMAT(2I4,20I2)                                                 FINPUT
      MNBAG(J) = NK                                                    FINPUT
      IF (NK.EQ.0) GO TO 68                                            FINPUT
      IF (IJK.EQ.0) WRITE (6,64)                                       FINPUT
   64 FORMAT(////5X,'AIRBAG',4X,'VS.',4X,'SEGMENTS',90X,'CARDS F.6')   FINPUT
      IF (K.NE.J) WRITE (6,42)                                         FINPUT
      IF (K.NE.J) STOP 20                                              FINPUT
      WRITE (6,65) J,(MBAG(2,I,J),MBAG(3,I,J),I=1,NK)                  FINPUT
   65 FORMAT('0     NO.',I2,12X,10(I3,'-',I3))                         FINPUT
      DO 66 I=1,NK                                                     FINPUT
      K = MBAG(2,I,J)                                                  FINPUT
   66 KTITLE(I) = SEG(K)                                               FINPUT
      WRITE (6,67) (BAGTTL(I,J),I=1,5),(KTITLE(I),I=1,NK)              FINPUT
   67 FORMAT(1X,5A4,10(3X,A4))                                         FINPUT
   68 CONTINUE                                                         FINPUT
C                                                                      FINPUT
C     INPUT CARDS F.7.A-F.7.B FOR SUBROUTINE WINDY.                    FINPUT
C                                                                      FINPUT
   69 DO  85  J=1,NGRND                                                FINPUT
   85 MWSEG(1,J) = 0                                                   FINPUT
      IF (NWINDF.EQ.0) GO TO 99                                        FINPUT
      READ  (5,33)  (MWSEG(1,J),J=1,NSEG)                              FINPUT
      IPAGE = 0                                                        FINPUT
      DO 73 J=1,NSEG                                                   FINPUT
      IWIND(J) = 0                                                     FINPUT
      WTIME(J) = 0.0                                                   FINPUT
      IF (MWSEG(1,J).EQ.0) GO TO 73                                    FINPUT
      IF (IPAGE.EQ.0) WRITE (6,70) NPG                                 PAGE
      IF (IPAGE.EQ.0) NPG=NPG+1                                        PAGE
   70 FORMAT('1 SEGMENT WIND FORCES',102X,'PAGE',I5/120X,'CARDS F.7'/  PAGE
     *       75X,'DRAG COEFFICIENT     BLOCKING'/                      WINDOP
     *       ' SEGMENT-ELLIPSOID   SEGMENT-PLANE',                     WINDOP
     *       16X,'WIND FORCE FUNCTION',10X,'FUNCTION',9X,              WINDOP
     *       'SEGMENTS-ELLIPSOID')                                     WINDOP
      IPAGE = 1                                                        FINPUT
```

```
      READ(5,86)(MWSEG(I,J),I=1,7),(MOWSEG(J,K),K=1,2*MWSEG(7,J))      WINDOP
   86 FORMAT (7I4,22I2/(I30,7I2))                                      WINDOP
      WRITE(6,71)(MWSEG(I,J),I=1,6)                                    OUT385
   71 FORMAT(1H0,I6,2H -,I3,I13,2H -,I3,I31,I23)                       OUT385
      IF (IABS(MWSEG(1,J)).NE.J) WRITE (6,42)                          WINDOP
      IF (IABS(MWSEG(1,J)).NE.J) STOP 21                               WINDOP
      M3 = MWSEG(3,J)                                                  FINPUT
      M4 = MWSEG(4,J)                                                  FINPUT
      M5 = MWSEG(5,J)                                                  FINPUT
      M6 = MWSEG(6,J)                                                  WINDOP
      M7 = MWSEG(7,J)                                                  OUT385
      DO 172 II=1,5                                                    FIXWBS
      KTITLE(II)=BLANK                                                 FIXWBS
  172 IF (M6.NE.0) KTITLE(II)=JTITLE(II,M6)                            FIXWBS
      WRITE (6,72) SEG(J),SEG(M3),(PLTTL(I,M4),I=1,5)                  FINPUT
     *               ,(JTITLE(I,M5),I=1,5),(KTITLE(I),I=1,5)           FIXWBS
     *               ,(MOWSEG(J,K),K=1,2*M7)                           OUT385
   72 FORMAT(3X,A4,14X,A4,1H-,5A4,3X,5A4,3X,5A4,2X,3(5(I3,1H-,I3)/94X)) OUT385
   73 CONTINUE                                                         FINPUT
   99 RETURN                                                           FINPUT
      END                                                              FINPUT
```

```
      SUBROUTINE FLXSEG                                                 FLXSEG
C                                                 REV IV    07/23/86TWOPI
      IMPLICIT REAL*8(A-H,O-Z)                                          FLXSEG
      COMMON/CONTRL/ TIME,NSEG,NJNT,NPL,NBLT,NBAG,NVEH,NGRND,           FLXSEG
     *               NS,NQ,NSD,NFLX,NHRNSS,NWINDF,NJNTF,NPRT(36),NPG    PAGE
      COMMON/SGMNTS/ D(3,3,30),WMEG(3,30),WMEGD(3,30),U1(3,30),U2(3,30),FLXSEG
     *               SEGLP(3,30),SEGLV(3,30),SEGLA(3,30),NSYM(30)       FLXSEG
      COMMON/FLXBLE/ HF(4,12,8),B42(3,3,24),V4(3,8),NFLEX(3,8)          FLXSEG
      COMMON/CNSNTS/ PI,RADIAN,G,THIRD,EPS(24),                         FLXSEG
     *               UNITL,UNITM,UNITT,GRAVTY(3),TWOPI                  TWOPI
      COMMON/TEMPVS/ TT(3,3), THN(4), CN1(3,3), CN(3,3), WNM1(3),       FLXSEG
     *               THND(4), PTD(3), WCSN(3), RHSN(3), RHS1(3),        FLXSEG
     *               RHS2(3), GF(3,4), GC(3,3), CGC(3,3), THA(3),       FLXSEG
     *              THAD(3), THADEG(3), DN2N1(3,3), RMG(3)              FLXSEG
      DIMENSION IDYPR(3)                                                FLXSEG
      DATA      IDYPR/3,2,1/                                            FLXSEG
      IF (NFLX.EQ.0) GO TO 99                                           FLXSEG
      CALL ELTIME(1,34)                                                 FLXSEG
      IFX = 1                                                           FLXSEG
   11 N1 = NFLEX(1,IFX)                                                 FLXSEG
      N3 = NFLEX(3,IFX)                                                 FLXSEG
      CALL DOTT33(D(1,1,N3),D(1,1,N1),TT)                               FLXSEG
      THN(1) =  DATAN2(TT(1,2),TT(1,1))                                 FLXSEG
      THN(2) = -DASIN(TT(1,3))                                          FLXSEG
      THN(3) =  DATAN2(TT(2,3),TT(3,3))                                 FLXSEG
      THN(4) =  1.0                                                     FLXSEG
      CT22 = 1.0-TT(1,3)**2                                             FLXSEG
      CT2  = DSQRT(CT22)                                                FLXSEG
      ST2  = -TT(1,3)                                                   FLXSEG
      CT1  =  TT(1,1)/CT2                                               FLXSEG
      ST1  =  TT(1,2)/CT2                                               FLXSEG
      CN1(1,1) = -TT(1,1)*TT(1,3)/CT22                                  FLXSEG
      CN1(1,2) = -TT(1,2)*TT(1,3)/CT22                                  FLXSEG
      CN1(1,3) =  1.0                                                   FLXSEG
      CN1(2,1) = -ST1                                                   FLXSEG
      CN1(2,2) =  CT1                                                   FLXSEG
      CN1(2,3) =  0.0                                                   FLXSEG
      CN1(3,1) =  TT(1,1)/CT22                                          FLXSEG
      CN1(3,2) =  TT(1,2)/CT22                                          FLXSEG
      CN1(3,3) =  0.0                                                   FLXSEG
      CALL DOT31(TT,WMEG(1,N3),WNM1)                                    FLXSEG
      DO 12 I=1,3                                                       FLXSEG
   12 WNM1(I) = WNM1(I) - WMEG(I,N1)                                    FLXSEG
      CALL MAT31(CN1,WNM1,THND)                                         FLXSEG
      THND(4) =  0.0                                                    FLXSEG
      CALL CROSS(WMEG(1,N1),WNM1,WCSN)                                  FLXSEG
      RHSN(1) = ( (-THND(1)*ST1*ST2 + THND(2)*CT1/CT2)*WNM1(1)          FLXSEG
     *            +( THND(1)*CT1*ST2 + THND(2)*ST1/CT2)*WNM1(2) )/CT2   FLXSEG
      RHSN(2) =  -THND(1)*(CT1*WNM1(1) + ST1*WNM1(2))                   FLXSEG
      RHSN(3) = ( (-THND(1)*ST1 + THND(2)*CT1*ST2/CT2)*WNM1(1)          FLXSEG
```

```
     *           +( THND(1)*CT1 + THND(2)*ST1*ST2/CT2)*WNM1(2) )/CT2    FLXSEG
   13 N2 = NFLEX(2,IFX)                                                 FLXSEG
      M = 0                                                             FLXSEG
      DO 15 I=1,3                                                       FLXSEG
      DO 14 J=1,4                                                       FLXSEG
      JM = J+M                                                          FLXSEG
      GF(I,J) = 0.0                                                     FLXSEG
      DO 14 K=1,4                                                       FLXSEG
   14 GF(I,J) = GF(I,J) + HF(K,JM,IFX)*THN(K)                           FLXSEG
   15 M = M+4                                                           FLXSEG
      DO 17 I=1,3                                                       FLXSEG
      THA(I)  = 0.0                                                     FLXSEG
      THAD(I) = 0.0                                                     FLXSEG
      DO 16 J=1,4                                                       FLXSEG
      THA (I) = THA (I) + GF(I,J)*THN (J)                               FLXSEG
   16 THAD(I) = THAD(I) + GF(I,J)*THND(J)                               FLXSEG
      THA (I) = 0.5*THA(I)                                              FLXSEG
   17 THADEG(I) = THA(I)/RADIAN                                         FLXSEG
      CALL DRCYPR (DN2N1,THADEG,IDYPR)                                  FLXSEG
      CALL MAT33(DN2N1,D(1,1,N1),D(1,1,N2))                             FLXSEG
      CSC = DCOS(THA(2))                                                FLXSEG
      CSS = DSIN(THA(2))                                                FLXSEG
      CN(1,1) =   0.0                                                   FLXSEG
      CN(2,1) =   0.0                                                   FLXSEG
      CN(3,1) =   1.0                                                   FLXSEG
      CN(1,2) =  -DSIN(THA(1))                                          FLXSEG
      CN(2,2) =   DCOS(THA(1))                                          FLXSEG
      CN(3,2) =   0.0                                                   FLXSEG
      CN(1,3) =   CSC*CN(2,2)                                           FLXSEG
      CN(2,3) =  -CSC*CN(1,2)                                           FLXSEG
      CN(3,3) =  -CSS                                                   FLXSEG
      CALL MAT33(GF, CN1, GC)                                           FLXSEG
      CALL MAT33(CN, GC, CGC)                                           FLXSEG
      CALL DOT33 (D(1,1,N1),CGC,B42(1,1,3*IFX-2))                       FLXSEG
      CALL DOTT33(B42(1,1,3*IFX-2),TT,B42(1,1,3*IFX))                   FLXSEG
      DO 20 I=1,3                                                       FLXSEG
      DO 20 J=1,3                                                       FLXSEG
      B42(I,J,3*IFX-2) = B42(I,J,3*IFX-2) - D(J,I,N1)                   FLXSEG
      B42(I,J,3*IFX-1) = D(J,I,N2)                                      FLXSEG
   20 B42(I,J,3*IFX  ) = -B42(I,J,3*IFX)                                FLXSEG
C                                                                       FLXSEG
C     COMPUTE V4                                                        FLXSEG
C                                                                       FLXSEG
      CALL MAT31(CGC,WNM1,RHS1)                                         FLXSEG
      DO 21 I=1,3                                                       FLXSEG
   21 RMG(I) = RHS1(I) + WMEG(I,N1)                                     FLXSEG
      CALL MAT31(DN2N1,RMG,WMEG(1,N2))                                  FLXSEG
      CALL CROSS(WMEG(1,N1),RHS1,RHS2)                                  FLXSEG
      CALL MAT31(CGC,WCSN,RHS1)                                         FLXSEG
      DO 25 I=1,3                                                       FLXSEG
```

```
25 RHS1(I) = RHS2(I) - RHS1(I)                                       FLXSEG
   CALL MAT31(GC,WNM1,RHS2)                                           FLXSEG
   RHS1(1) = RHS1(1) - THAD(1)*(CN(2,2)*RHS2(2)-CN(1,2)*CSC*RHS2(3)) FLXSEG
  *                  - THAD(2)*CN(2,2)*CSS*RHS2(3)                    FLXSEG
   RHS1(2) = RHS1(2) + THAD(1)*(CN(1,2)*RHS2(2)+CN(2,2)*CSC*RHS2(3)) FLXSEG
  *                  + THAD(2)*CN(1,2)*CSS*RHS2(3)                    FLXSEG
   RHS1(3) = RHS1(3) - THAD(2)*CSC*RHS2(3)                            FLXSEG
   CALL MAT31(GF, RHSN, RHS2)                                         FLXSEG
   M = 1                                                              FLXSEG
   DO 30 I=1,3                                                        FLXSEG
   DO  26  J=1,3                                                      FLXSEG
   PTD(J) = 0.0                                                       FLXSEG
   DO  26  K=1,3                                                      FLXSEG
   KK = K+M-1                                                         FLXSEG
26 PTD(J) = PTD(J) + HF(J,KK,IFX)*THND(K)                             FLXSEG
   RHS2(I) = RHS2(I) + XDY(PTD,CN1,WNM1)                              FLXSEG
30 M = M+4                                                            FLXSEG
   CALL MAT31(CN, RHS2, PTD)                                          FLXSEG
   DO 35 I=1,3                                                        FLXSEG
35 RHS1(I) = RHS1(I) + PTD(I)                                         FLXSEG
   CALL DOT31(D(1,1,N1),RHS1,V4(1,IFX))                               FLXSEG
   IF (IFX.EQ.NFLX) GO TO 98                                          FLXSEG
   IFX = IFX+1                                                        FLXSEG
   IF (NFLEX(1,IFX).EQ.N1 .AND. NFLEX(3,IFX).EQ.N3) GO TO 13          FLXSEG
   GO TO 11                                                           FLXSEG
98 CALL ELTIME(2,34)                                                  FLXSEG
99 RETURN                                                             FLXSEG
   END                                                                FLXSEG
```

```
      DOUBLE PRECISION FUNCTION FNTERP(THETA,PHI,NT)                    FNTERP
C                                                  REV IV    04/10/87FNFIX
C     COMPUTES THE RESTORING TORQUE OF A JOINT AS A FUNCTION OF THE     FNTERP
C     FLEXURE ANGLE (THETA) AND THE AZIMUTH ANGLE (PHI) AS DEFINED BY   FNTERP
C     FUNCTION NO. NT                                                   FNTERP
C                                                                       FNTERP
C     ASSUMES  0  < THETA < PI                                          FNTERP
C             -PI <  PHI  < PI                                          FNTERP
C              DATA IN TAB ARRAY CONTAINS NTHETA,NPHI FOLLOWED BY       FNTERP
C              TWO DIMENSIONAL ARRAY OF FUNCTIONAL VALUES (NTHETA > 0)  FNTERP
C              OR POLYNOMIAL COEFFICIENTS (NTHETA < 0) FOR EQUALLY      FNTERP
C              SPACED VALUES OF PHI.                                    FNTERP
C                                                                       FNTERP
C                   THETA(I) = (I-1)*PI/(NTHETA-1) FOR I=1,NTHETA       FNTERP
C                   PHI(J) = -PI + (J-1)*2*PI/NPHI FOR J=1,NPHI         FNTERP
C                       F(THETA,PI) = F(THETA,-PI)                      FNTERP
C                                                                       FNTERP
C     SUBROUTINE EVALUATES G1(THETA) = F(THETA,PHI(J)  )                FNTERP
C                          G2(THETA) = F(THETA,PHI(J+1))                FNTERP
C               FOR  PHI(J) < PHI < PHI(J+1)                            FNTERP
C     BY LINEAR INTERPOLATION OR POLYNOMIAL EVALUATION AND THEN LINEAR  FNTERP
C     INTERPOLATES BETWEEN G1 AND G2 TO OBTAIN F(THETA,PHI).            FNTERP
C     IF F < 0, F IS SET TO ZERO, THEREFORE A DEAD BAND IS OBTAINED     FNTERP
C     BY NEGATIVE VALUES IN THE TABLE.                                  FNTERP
C                                                                       FNTERP
      IMPLICIT REAL*8 (A-H,O-Z)                                         FNTERP
      COMMON/CNSNTS/ PI,RADIAN,G,THIRD,EPS(24),                         FNTERP
     *               UNITL,UNITM,UNITT,GRAVTY(3),TWOPI                  TWOPI
      COMMON/TABLES/MXNTI,MXNTB,MXTB1,MXTB2,NTI(50),NTAB(1250),TAB(4500)DIMENB
      IERROR = 0                                                        FNTERP
      IF (PHI.LT.-PI)   IERROR = 1                                      FNTERP
      IF (PHI.GT. PI)   IERROR = 2                                      FNTERP
      IF (THETA.LT.0.0) IERROR = 3                                      FNTERP
      IF (THETA.GT.PI ) IERROR = 4                                      FNTERP
      IF (IERROR.NE.0)  WRITE (6,11) IERROR,THETA,PHI,NT                FNTERP
   11 FORMAT('0 IMPROPER ARGUMENTS TO FUNCTION FNTERP. ERROR CODE =',I4/FNTERP
     *       '0 THETA =',G25.15, ' PHI =',G25.15,' NT =',I6)            FNTERP
      IF (IERROR.NE.0) STOP 36                                          FNTERP
      NF = NTI(NT) + 5                                                  FNTERP
      NTHETA = TAB(NF)                                                  FNTERP
      NPHI   = TAB(NF+1)                                                FNTERP
C                                                                       FNTERP
C     DETERMINE INDEX AND INTERPOLATION PARAMETERS FOR PHI.             FNTERP
C                                                                       FNTERP
      IF (PHI.GE.PI-EPS(15)) PHI=0.0-PI                                 FNFIX
      XNP = (PHI+PI)/TWOPI*TAB(NF+1)                                    TWOPI
      NP1 = XNP                                                         FNTERP
      NP2 = NP1+1                                                       FNTERP
      IF (NP2.GE.NPHI) NP2 = 0                                          FNTERP
      RP2 = XNP - DFLOAT(NP1)                                           FNTERP
```

```
      RP1 = 1.0 - RP2                                                   FNTERP
      NTH = IABS(NTHETA)                                                FNTERP
      IP1 = NF+1+NP1*NTH                                                FNTERP
      IP2 = NF+1+NP2*NTH                                                FNTERP
C                                                                       FNTERP
C     DETERMINE INDEX AND INTERPOLATION PARAMETERS FOR THETA.           FNTERP
C                                                                       FNTERP
      IF (NTHETA.LT.0) GO TO 20                                         FNTERP
      XNT = THETA/PI*(TAB(NF)-1.0)                                      FNTERP
      NT1 = XNT                                                         FNTERP
      RT2 = XNT - DFLOAT(NT1)                                           FNTERP
      RT1 = 1.0 - RT2                                                   FNTERP
      IT1 = IP1 + NT1                                                   FNTERP
      IT2 = IP2 + NT1                                                   FNTERP
      G1  = RT1*TAB(IT1+1) + RT2*TAB(IT1+2)                             FNTERP
      G2  = RT1*TAB(IT2+1) + RT2*TAB(IT2+2)                             FNTERP
      GO TO 23                                                          FNTERP
C                                                                       FNTERP
C     COMPUTE FOR POLYNOMIALS IN THETA FOR FIXED PHI.                   FNTERP
C                                                                       FNTERP
   20 NPOLY = -NTHETA-1                                                 FNTERP
      IT1 = IP1 + NPOLY + 2                                             FNTERP
      IT2 = IP2 + NPOLY + 2                                             FNTERP
      THETA1 = THETA - TAB(IP1+1)                                       FNTERP
      THETA2 = THETA - TAB(IP2+1)                                       FNTERP
      G1 = 0.0                                                          FNTERP
      G2 = 0.0                                                          FNTERP
      DO 21 I=1,NPOLY                                                   FNTERP
      IT1 = IT1-1                                                       FNTERP
      IT2 = IT2-1                                                       FNTERP
      G1 = THETA1*(TAB(IT1)+G1)                                         FNTERP
   21 G2 = THETA2*(TAB(IT2)+G2)                                         FNTERP
      IF (THETA1.LT.0.0) G1=0.0                                         FNFIX
      IF (THETA2.LT.0.0) G2=0.0                                         FNFIX
   23 FNTERP = RP1*G1 + RP2*G2                                          FNTERP
      IF (FNTERP.LT.0.0) FNTERP = 0.0                                   FNTERP
      RETURN                                                            FNTERP
      END                                                               FNTERP
```

```
      SUBROUTINE FRCDFL (D,RATE,M,N,FRCDF,ELOSS)                        FRCDFL
C                                                 REV III.2 08/08/84REVIII
C     EVALUATE FORCE DEFLECTION FUNCTION AT POINT D, WHERE DEFINITION   FRCDFL
C     OF FUNCTION IS CONTROLLED BY M INDEX OF NTAB ARRAY.               FRCDFL
C     DERIVATIVE, FUNCTION OR INTEGRAL IS EVALUATED AS N = 0,1 OR 2.    FRCDFL
C        NTAB(M)    -  INDEX TO TAB ARRAY FOR REAL DATA                 FRCDFL
C        NTAB(M+1)  -  INDEX TO TAB ARRAY FOR BASE FUNCTION             FRCDFL
C        NTAB(M+2)  -  INDEX TO TAB ARRAY FOR INERTIAL FUNCTION, IF ANY FRCDFL
C                                                                       FRCDFL
C     ASSUMES  0 < DG < DCUBIC < DREF < DMAX                            FRCDFL
C               BUT ANY < MAY BE LESS THAN OR EQUAL TO                  FRCDFL
C                                                                       FRCDFL
      IMPLICIT REAL*8(A-H,O-Z)                                          FRCDFL
      COMMON/TABLES/MXNTI,MXNTB,MXTB1,MXTB2,NTI(50),NTAB(1250),TAB(4500)DIMENB
      F = 0.0                                                           FRCDFL
      ELOSS = 0.0                                                       FRCDFL
      L = NTAB(M)                                                       FRCDFL
      TAB(L) = D                                                        FRCDFL
      IF (D.LT.0.0) GO TO 99                                            FRCDFL
      DMAX = TAB(L+8)                                                   FRCDFL
      IF (D.LT.DMAX) GO TO 10                                           FRCDFL
C                                                                       FRCDFL
C     DMAX < D , USE MAX VALUE                                          FRCDFL
C                                                                       FRCDFL
      IF (N-1)  99,9,99                                                 FRCDFL
    9 FDMAX = TAB(L+10)                                                 FRCDFL
      F = FDMAX                                                         FRCDFL
      GO TO 40                                                          FRCDFL
   10 DREF = TAB(L+7)                                                   FRCDFL
      IF (D.GE.DREF) GO TO 30                                           FRCDFL
      DCUBIC = TAB(L+6)                                                 FRCDFL
      IF (DCUBIC.GE.DREF) GO TO 20                                      FRCDFL
      IF (D.LE.DCUBIC) GO TO 20                                         FRCDFL
C                                                                       FRCDFL
C     DCUBIC < D < DREF , USE CUBIC                                     FRCDFL
C                                                                       FRCDFL
      LC = L+14                                                         FRCDFL
      DC0 = TAB(L+18)                                                   FRCDFL
      X = D-DC0                                                         FRCDFL
      IF (N-1) 12,11,99                                                 FRCDFL
C                                                                       FRCDFL
C     USE CUBIC DEFINITION                                              FRCDFL
C                                                                       FRCDFL
   11  F = TAB(LC) + X *(TAB(LC+1)+X*(TAB(LC+2)+X*TAB(LC+3)))           FRCDFL
      GO TO 40                                                          FRCDFL
C                                                                       FRCDFL
C     USE DERIVATIVE OF CUBIC                                           FRCDFL
C                                                                       FRCDFL
   12 F = TAB(LC+1)+X*(2.0*TAB(LC+2)+X*3.0*TAB(LC+3))                   FRCDFL
      GO TO 99                                                          FRCDFL
```

```
   20 DG = TAB(L+5)                                                    FRCDFL
      IF (D.LE.DG) GO TO 40                                            FRCDFL
C                                                                      FRCDFL
C     DG < D < DCUBIC , USE QUADRATIC                                  FRCDFL
C                                                                      FRCDFL
      LQ = L+11                                                        FRCDFL
      X = D-DG                                                         FRCDFL
      IF (N-1) 22,21,99                                                FRCDFL
C                                                                      FRCDFL
C     USE QUADRATIC DEFINITION                                         FRCDFL
C                                                                      FRCDFL
  21  F = TAB(LQ)+X*(TAB(LQ+1)+X*TAB(LQ+2))                            FRCDFL
      GO TO 40                                                         FRCDFL
C                                                                      FRCDFL
C     USE DERIVATIVE OF QUADRATIC.                                     FRCDFL
C                                                                      FRCDFL
  22  F = TAB(LQ+1)+X*2.0*TAB(LQ+2)                                    FRCDFL
      GO TO 99                                                         FRCDFL
C                                                                      FRCDFL
C     DREF < D < DMAX, USE BASE FUNCTION                               FRCDFL
C                                                                      FRCDFL
  30  IF (N-1) 31,31,99                                                FRCDFL
  31  NB = NTAB(M+1)                                                   FRCDFL
C                                                                      FRCDFL
C     EVALUATE BASE FUNCTION                                           FRCDFL
C                                                                      FRCDFL
      IF (NB.GT.0) F = EVALFD(D,NB,N)                                  FRCDFL
      NI = NTAB(M+2)                                                   FRCDFL
C                                                                      FRCDFL
C     ADD INERTIAL FUNCTION , IF ANY                                   FRCDFL
C                                                                      FRCDFL
      IF (NI.GT.0) F = F+EVALFD(D,NI,N)                                FRCDFL
   40 IF  (N.NE.1)  GO TO 99                                           FRCDFL
C                                                                      FRCDFL
C     COMPUTE AND ADD RATE DEPENDENT FUNCTIONS, IF ANY.                FRCDFL
C                                                                      FRCDFL
C     CURRENT RESTRICTIONS:                                            FRCDFL
C                                                                      FRCDFL
C        1) COMPUTED FOR N=1 (FUNCTION) ONLY.                          FRCDFL
C                                                                      FRCDFL
C        2) FUNCTION NOS. M+2,M+3 AND M+4 (USED FOR INERTIAL SPIKE,    FRCDFL
C           R FACTOR AND G FACTOR FUNCTIONS) MUST BE NEGATIVE OR ZERO, FRCDFL
C           I.E., THESE FUNCTIONS CANNOT BE USED IN CONJUNCTION WITH   FRCDFL
C           THE RATE DEPENDENT FUNCTIONS.                              FRCDFL
C                                                                      FRCDFL
C        3) ASSUMES THE FUNCTIONAL FORM                                FRCDFL
C                                                                      FRCDFL
C                 F(D,D') = F1(D) + F2(D)*F3(D') + F4(D')              FRCDFL
C                                                                      FRCDFL
C           WHERE F1(D ) IS DEFINED BY FUNCTION NTAB(M+1)>0,           FRCDFL
```

```
C                    I.E., NORMAL FORCE DEFLECTION FUNCTION WITH NO FRCDFL
C                    INERTIAL SPIKE FUNCTION AND DEFAULT VALUES     FRCDFL
C                    R=1 AND G=0 (UNLOADING AND RELOADING SAME AS   FRCDFL
C                    ORIGINAL LOADING);                             FRCDFL
C                                                                   FRCDFL
C             F2(D ) IS DEFINED BY FUNCTION NTAB(M+2)<0,            FRCDFL
C                    IF NTAB(M+2)=0, F2(D )=0;                      FRCDFL
C                                                                   FRCDFL
C             F3(D') IS DEFINED BY FUNCTION NTAB(M+3)<0,            FRCDFL
C                    IF NTAB(M+3)=0, F3(D')=0;                      FRCDFL
C                                                                   FRCDFL
C         AND F4(D') IS DEFINED BY FUNCTION NTAB(M+4)<0,            FRCDFL
C                    IF NTAB(M+4)=0, F4(D')=0.                      FRCDFL
C                                                                   FRCDFL
C         NOTE: FUNCTIONAL FORM CAN BE CHANGED BY REVISING PROGRAM  FRCDFL
C         BETWEEN STATEMENTS 40 AND 99.                             FRCDFL
C                                                                   FRCDFL
      F2 = 0.0                                                      FRCDFL
      F3 = 0.0                                                      FRCDFL
      F4 = 0.0                                                      FRCDFL
      N2 = -NTAB(M+2)                                               FRCDFL
      N3 = -NTAB(M+3)                                               FRCDFL
      N4 = -NTAB(M+4)                                               FRCDFL
      IF  (N2.GT.0)  F2 = EVALFD (D,   N2,N)                        FRCDFL
      IF  (N3.GT.0)  F3 = EVALFD (RATE,N3,N)                        FRCDFL
      IF  (N4.GT.0)  F4 = EVALFD (RATE,N4,N)                        FRCDFL
      F = F + F2*F3 + F4                                            FRCDFL
      ELOSS = RATE*(F2*F3+F4)                                       FRCDFL
   99 FRCDF = F                                                     FRCDFL
      RETURN                                                        FRCDFL
      END                                                           FRCDFL
```

```
      SUBROUTINE FSMSOL (C,R,NN,MX,MAXN,JN,MAXDIM)                      FSMSOL
C                                                 REV III.2 08/08/84REVIII
C        SOLVES A SET OF SIMULTANEOUS EQUATIONS  OF SIZE 3*MM           FSMSOL
C        WHERE THE MATRIX CONSISTS OF A SET OF 3*3 SUBMATRICES          FSMSOL
C        STORED IN  C(3,3,IJ). THE LOCATION OF THE I,J ELEMENT          FSMSOL
C        IS  STORED IN   NN(I,J).  I.E.  IJ= NN(I,J)                    FSMSOL
C                                                                       FSMSOL
C        A NEGATIVE IJ IMPLIES THAT C( , ,!IJ!) IS AN                   FSMSOL
C        IDENTITY AND THE RIGHT SIDE IS ZERO. A NEGATIVE                FSMSOL
C        IJ WILL ONLY OCCUR ON A DIAGONAL ENTRY OF NN.                  FSMSOL
C                                                                       FSMSOL
C            THE  BASIC EQUATION IS   CX=R                              FSMSOL
C                                                                       FSMSOL
C           DURING THE SOLUTION THE C MATRIX IS DESTROYED ,IT MAY       FSMSOL
C           BE NECESSARY TO ADD TO THE C ARRAY.                         FSMSOL
C           THE SOLUTION IS STORED IN R.                                FSMSOL
C                                                                       FSMSOL
C       INPUT                                                           FSMSOL
C                                                                       FSMSOL
C           C(3,3,K)  GIVEN ARRAY                                       FSMSOL
C           R(3,MM)   GIVEN RIGHT HAND SIDE                             FSMSOL
C           NN(JJ,JJ) GIVEN ARRAY CONTAINING LOCATIONS OF I,J,ELEMENT   FSMSOL
C           MX        SIZE OF SYSTEM OF SUBMATRICES (POSITIVE INDICATES FSMSOL
C                     THAT C MATRIX IS SYMMETRIC, NEGATIVE IT IS NOT.)  FSMSOL
C           MAXN      LARGEST VALUE IN NN ARRAY                         FSMSOL
C           JN        DIMENSION OF NN                                   FSMSOL
C           MAXDIM    THIRD DIMENSION OF C IN CALLING ROUTINE           FSMSOL
C                                                                       FSMSOL
      IMPLICIT REAL*8 (A-H,O-Z)                                         FSMSOL
      COMMON/CONTRL/ TIME,NSEG,NJNT,NPL,NBLT,NBAG,NVEH,NGRND,           PAGE
     *               NS,NQ,NSD,NFLX,NHRNSS,NWINDF,NJNTF,NPRT(36),NPG    PAGE
      DIMENSION C(3,3,1),R(3,1),NN(JN,1)                                FSMSOL
      CALL ELTIME(1,20)                                                 FSMSOL
      MM = IABS(MX)                                                     FSMSOL
      IF  (MM.LE.0)  GO TO 99                                           FSMSOL
      MM1 = MM-1                                                        FSMSOL
      MP1 = MM+1                                                        FSMSOL
      DO  50  II=1,MM                                                   FSMSOL
      I = MP1-II                                                        FSMSOL
C                                                                       FSMSOL
C     START PIVOT AT BOTTOM - FIND PIVOT - INVERT.                      FSMSOL
C                                                                       FSMSOL
      L = NN(I,I)                                                       FSMSOL
      IF  (L.LE.0)  GO TO 50                                            FSMSOL
      DO  14  M=1,3                                                     FSMSOL
      B = 1.0/C(M,M,L)                                                  FSMSOL
      C(M,M,L) = 1.0                                                    FSMSOL
      C(M,1,L) = B*C(M,1,L)                                             FSMSOL
      C(M,2,L) = B*C(M,2,L)                                             FSMSOL
      C(M,3,L) = B*C(M,3,L)                                             FSMSOL
```

```
      R(M,I)    = B*R(M,I)                                          FSMSOL
      DO  13  N=1,3                                                  FSMSOL
      IF  (N.EQ.M)  GO TO 13                                         FSMSOL
      B = C(N,M,L)                                                   FSMSOL
      C(N,M,L) = 0.0                                                 FSMSOL
      C(N,1,L) = C(N,1,L) - B*C(M,1,L)                               FSMSOL
      C(N,2,L) = C(N,2,L) - B*C(M,2,L)                               FSMSOL
      C(N,3,L) = C(N,3,L) - B*C(M,3,L)                               FSMSOL
      R(N,I)   = R(N,I)   - B*R(M,I)                                 FSMSOL
   13 CONTINUE                                                       FSMSOL
   14 CONTINUE                                                       FSMSOL
C                                                                    FSMSOL
C     CHECK IF DONE.                                                 FSMSOL
C                                                                    FSMSOL
      IF  (I.EQ.1)  GO TO 50                                         FSMSOL
      IM1 = I-1                                                      FSMSOL
C                                                                    FSMSOL
C     CALCULATE PIVOT ROW.                                           FSMSOL
C                                                                    FSMSOL
      DO  20  J=1,IM1                                                FSMSOL
      IF  (NN(I,J).EQ.0)  GO TO 20                                   FSMSOL
      M = NN(I,J)                                                    FSMSOL
      DO  15  N=1,3                                                  FSMSOL
      A = C(1,1,L)*C(1,N,M) + C(1,2,L)*C(2,N,M) + C(1,3,L)*C(3,N,M)  FSMSOL
      B = C(2,1,L)*C(1,N,M) + C(2,2,L)*C(2,N,M) + C(2,3,L)*C(3,N,M)  FSMSOL
      D = C(3,1,L)*C(1,N,M) + C(3,2,L)*C(2,N,M) + C(3,3,L)*C(3,N,M)  FSMSOL
      C(1,N,M) = A                                                   FSMSOL
      C(2,N,M) = B                                                   FSMSOL
   15 C(3,N,M) = D                                                   FSMSOL
   20 CONTINUE                                                       FSMSOL
C                                                                    FSMSOL
C     DONE WITH PIVOT ROW - ZERO COLUMN I ABOVE DIAGONAL.            FSMSOL
C                                                                    FSMSOL
C     1,1                                                            FSMSOL
C         .                                                          FSMSOL
C             .                                                      FSMSOL
C              K,K .  . K,J .  . K,I          C   = C   - C  *C      FSMSOL
C                .  .    .        .            KJ    KJ    KI  IJ    FSMSOL
C                .     .  .                                          FSMSOL
C              J,K .  . J,J .  . J,I          C   = C   - C  *C      FSMSOL
C                .       .  .     .            JK    JK    JI  IK    FSMSOL
C                .       .    .  .                                   FSMSOL
C              I,K .  . I,J .  . I,I          C   = 0                FSMSOL
C                                  .           KI                    FSMSOL
C                                      .                             FSMSOL
C                                        M,M                         FSMSOL
C                                                                    FSMSOL
      DO  40  K=1,IM1                                                FSMSOL
      KI = NN(K,I)                                                   FSMSOL
      IK = NN(I,K)                                                   FSMSOL
```

```
      IF  (KI.EQ.0 .AND. IK.EQ.0)  GO TO 40                            FSMSOL
      DO  30  J=K,IM1                                                   FSMSOL
      IJ = NN(I,J)                                                      FSMSOL
      JI = NN(J,I)                                                      FSMSOL
      IF  (KI.EQ.0  .OR. IJ.EQ.0)  GO TO 24                             FSMSOL
      KJ = NN(K,J)                                                      FSMSOL
      IF  (KJ.NE.0)  GO TO 22                                           FSMSOL
      MAXN = MAXN+1                                                     FSMSOL
      IF  (MAXN.GT.MAXDIM)  GO TO 41                                    FSMSOL
      KJ = MAXN                                                         FSMSOL
      NN(K,J) = KJ                                                      FSMSOL
      DO  21  M=1,3                                                     FSMSOL
      DO  21  N=1,3                                                     FSMSOL
   21 C(N,M,KJ) = 0.0                                                   FSMSOL
   22 DO  23  M=1,3                                                     FSMSOL
      DO  23  N=1,3                                                     FSMSOL
   23 C(N,M,KJ) = C(N,M,KJ) - C(N,1,KI)*C(1,M,IJ)                       FSMSOL
     *                      - C(N,2,KI)*C(2,M,IJ)                       FSMSOL
     *                      - C(N,3,KI)*C(3,M,IJ)                       FSMSOL
   24 IF  (J.EQ.K)  GO TO 30                                            FSMSOL
      IF  (JI.EQ.0  .OR. IK.EQ.0)  GO TO 30                             FSMSOL
      JK = NN(J,K)                                                      FSMSOL
      IF  (JK.NE.0)  GO TO 26                                           FSMSOL
      MAXN = MAXN+1                                                     FSMSOL
      IF  (MAXN.GT.MAXDIM)  GO TO 41                                    FSMSOL
      JK = MAXN                                                         FSMSOL
      NN(J,K) = JK                                                      FSMSOL
      DO  25  M=1,3                                                     FSMSOL
      DO  25  N=1,3                                                     FSMSOL
   25 C(N,M,JK) = 0.0                                                   FSMSOL
   26 IF  (MX.LT.0)  GO TO 28                                           FSMSOL
      DO  27  M=1,3                                                     FSMSOL
      DO  27  N=1,3                                                     FSMSOL
   27 C(N,M,JK) = C(M,N,KJ)                                             FSMSOL
      GO TO 30                                                          FSMSOL
   28 DO  29  M=1,3                                                     FSMSOL
      DO  29  N=1,3                                                     FSMSOL
   29 C(N,M,JK) = C(N,M,JK) - C(N,1,JI)*C(1,M,IK)                       FSMSOL
     *                      - C(N,2,JI)*C(2,M,IK)                       FSMSOL
     *                      - C(N,3,JI)*C(3,M,IK)                       FSMSOL
   30 CONTINUE                                                          FSMSOL
      IF  (KI.EQ.0)  GO TO 40                                           FSMSOL
      DO  35  N=1,3                                                     FSMSOL
   35 R(N,K) = R(N,K) - C(N,1,KI)*R(1,I)                                FSMSOL
     *                - C(N,2,KI)*R(2,I)                                FSMSOL
     *                - C(N,3,KI)*R(3,I)                                FSMSOL
   40 CONTINUE                                                          FSMSOL
   50 CONTINUE                                                          FSMSOL
      GO TO 51                                                          FSMSOL
   41 WRITE  (6,49)  MAXDIM,NPG,(L,L=1,MM)                              PAGE
```

```
      NPG=NPG+1                                                           PAGE
      DO  42  I=1,MM                                                      FSMSOL
   42 WRITE  (6,43)  I,(NN(I,L),L=1,MM)                                   FSMSOL
   43 FORMAT(I3,3X,40I3,3X/6X,40I3)                                       FSMSOL
      WRITE  (6,44) NPG                                                   PAGE
      NPG=NPG+1                                                           PAGE
   44 FORMAT('1 FSMSOL PRINT OF RHS ARRAY',96X,'PAGE',I5//)               PAGE
      DO  45  K=1,MM                                                      FSMSOL
   45 WRITE  (6,46)  K,(R(I,K),I=1,3)                                     FSMSOL
   46 FORMAT(I6,9G14.7)                                                   FSMSOL
      WRITE  (6,47) NPG                                                   PAGE
      NPG=NPG+1                                                           PAGE
   47 FORMAT('1 FSMSOL PRINT OF C ARRAY ELEMENTS',89X,'PAGE',I5//)        PAGE
      DO  48  K=1,MAXN                                                    FSMSOL
   48 WRITE  (6,46)  K,((C(I,L,K),L=1,3),I=1,3)                           FSMSOL
   49 FORMAT('1 MAXIMUM DIMENSION OF',I4,' ON C ARRAY HAS BEEN EXCEEDED FSMSOL
     *IN SUBROUTINE FSMSOL.',46X,'PAGE',I5//' IF 600, CALL IS FROM SUBROPAGE
     *UTINE DAUX. IF 200'                                                 PAGE
     * ,' CALL IS FROM SUBROUTINE HPTURB.'//' PROGRAM IS BEING TERMINATEPAGE
     *D. COMPLETE PRINT-OUT OF IJK, RHS AND C ARRAYS FOLLOW.'//           FSMSOL
     *' FSMSOL PRINT OF IJK MATRIX'//(6X,40I3))                           FSMSOL
      STOP 35                                                             FSMSOL
C                                                                         FSMSOL
C     BACKDOWN SOLUTION                                                   FSMSOL
C                                                                         FSMSOL
   51 IF  (MM.EQ.1)  GO TO 99                                             FSMSOL
      DO  90  J=1,MM1                                                     FSMSOL
      IP = J+1                                                            FSMSOL
      DO  80  I=IP,MM                                                     FSMSOL
      IF  (NN(I,J).EQ.0)  GO TO 80                                        FSMSOL
      IJ = NN(I,J)                                                        FSMSOL
      DO  75  N=1,3                                                       FSMSOL
   75 R(N,I) = R(N,I) - C(N,1,IJ)*R(1,J)                                  FSMSOL
     *                - C(N,2,IJ)*R(2,J)                                  FSMSOL
     *                - C(N,3,IJ)*R(3,J)                                  FSMSOL
   80 CONTINUE                                                            FSMSOL
   90 CONTINUE                                                            FSMSOL
   99 CALL ELTIME(2,20)                                                   FSMSOL
      RETURN                                                              FSMSOL
      END                                                                 FSMSOL
```

```
      SUBROUTINE GLOBAL (J,HD3,DH1,TQC,T9,ANGL)                         GLOBAL
C                                                REV IV    07/24/86SLIP
      IMPLICIT REAL*8 (A-H,O-Z)                                         GLOBAL
      DIMENSION HD3(3),DH1(3,3),T9(3),ANGL(3),CC(3)                     GLOBAL
      COMMON/DESCRP/ PHI(3,30),W(30),RW(30),SR(4,60),HA(3,60),HB(3,60), SLIP
     *               RPHI(3,30),HT(3,3,60),SPRING(5,90),VISC(7,90),     GLOBAL
     *               JNT(30),IPIN(30),ISING(30),IGLOB(30),JOINTF(30)    GLOBAL
      COMMON/TABLES/MXNTI,MXNTB,MXTB1,MXTB2,NTI(50),NTAB(1250),TAB(4500)DIMENB
      COMMON/TEMPVI/ CREST,TTI(3),R1I(3),R2I(3),JSTOP(4,2,30)           GLOBAL
      COMMON/CNSNTS/ PI,RADIAN,G,THIRD,EPS(24),                         GLOBAL
     *               UNITL,UNITM,UNITT,GRAVTY(3),TWOPI                  TWOPI
      IF (DABS(HD3(3)).GT.1.0-EPS(6)) GO TO 34                          GLOBAL
      ANGL(1) = DACOS(HD3(3))                                           GLOBAL
      NT = IGLOB(J)                                                     GLOBAL
      NT1 = NTAB(NT+2)                                                  GLOBAL
      CALL HERRON(HD3,NT1,THETO,THETOP)                                 GLOBAL
      JSTOP(4,1,J) = 0                                                  GLOBAL
      IF (ANGL(1).LE.THETO) GO TO 34                                    GLOBAL
      JSTOP(4,1,J) = 1                                                  GLOBAL
      MT = NTAB(NT+5)                                                   GLOBAL
      CREST = TAB(MT+3)                                                 GLOBAL
      STH2 = 1.0-HD3(3)**2                                              GLOBAL
      STH = DSQRT(STH2)                                                 GLOBAL
      CTH = HD3(3)/STH                                                  GLOBAL
      CST = DSQRT(STH2+THETOP**2)                                       GLOBAL
      DR  = (ANGL(1)-THETO)*STH/CST                                     GLOBAL
      LT  = NTAB(NT)                                                    GLOBAL
      TAB(LT) = DR                                                      GLOBAL
      NTAB(NT+2) = 0                                                    GLOBAL
      DRDOT = 0.0                                                       GLOBAL
      CALL FRCDFL (DR,DRDOT,NT,1,TQF,ELOSS)                             GLOBAL
      NTAB(NT+2) = NT1                                                  GLOBAL
      TQC = TQF/CST                                                     GLOBAL
      CC(1) = -HD3(2)+HD3(1)*CTH*THETOP                                 GLOBAL
      CC(2) =  HD3(1)+HD3(2)*CTH*THETOP                                 GLOBAL
      CC(3) =                -STH*THETOP                                GLOBAL
      DO 28 L=1,3                                                       GLOBAL
   28 T9(L) = CC(1)*DH1(L,1) + CC(2)*DH1(L,2) + CC(3)*DH1(L,3)          GLOBAL
   34 RETURN                                                            GLOBAL
      END                                                               GLOBAL
```

```
      SUBROUTINE HBELT (J1,J2,KNLO,IND)                                 HBELT
C                                                  REV IV    02/01/88MISDOT
C     ARGUMENTS:                                                        HBELT
C         J1,J2 - FIRST AND LAST INDEX FOR BELTS.                       HBELT
C          KNLO - ZERO VALUE FOR KNL INDEX.                             HBELT
C          IND  - 0: CALL IS FROM SUBROUTINE CONTCT                     HBELT
C                 1: CALL IS FROM SUBROUTINE UPDATE                     HBELT
C                                                                       HBELT
      IMPLICIT REAL*8 (A-H,O-Z)                                         HBELT
      COMMON/CNTSRF/ PL(24,30),BELT(20,8),TPTS(6,8),BD(24,40)           EDGE
      COMMON/CNSNTS/ PI,RADIAN,G,THIRD,EPS(24),                         MISDOT
     *               UNITL,UNITM,UNITT,GRAVTY(3),TWOPI                  MISDOT
      COMMON/SGMNTS/ D(3,3,30),WMEG(3,30),WMEGD(3,30),U1(3,30),U2(3,30),HBELT
     *               SEGLP(3,30),SEGLV(3,30),SEGLA(3,30),NSYM(30)       HBELT
      COMMON/TABLES/MXNTI,MXNTB,MXTB1,MXTB2,NTI(50),NTAB(1250),TAB(4500)DIMENB
      COMMON/FORCES/PSF(7,70),BSF(4,20),SSF(10,40),BAGSF(3,20),         NCFORC
     *               PRJNT(7,30),NPANEL(5),NPSF,NBSF,NSSF,NBGSF         HBELT
      COMMON/HRNESS/ BAR(15,100),BB(100),BBDOT(100),PLOSS(2,100),       HBELT
     *               XLONG(20),HTIME(2),IBAR(5,100),NL(2,100),          HBELT
     *               NPTSPB(20),NPTPLY(20),NTHRNS(20),NBLTPH(5)         HBELT
C     THIS COMMON/TEMPVS/ IS SHARED BY HPTURB, HBPLAY, HBELT AND HSETC. HBELT
      COMMON/TEMPVS/ B(3,3,3),S(3,3),T(3),R(3),V(3),T1(3),T2(3),        HBELT
     *               E(3,3,50),EDOT(3,50),FCE(3,50),FR(3,50),ZR(3,50),  HBELT
     *               TR(3,50),U(3,50),PTLOSS(2,50),BL(50),FB(50),FP(50),HBELT
     *               OLDBB(100),RHS(3,54),C(3,3,200),IJK(54,54)         HBELT
      CALL ELTIME (1,38)                                                HBELT
      NTP = 0                                                           HBELT
      K2 = 0                                                            HBELT
      DO  31  JB=J1,J2                                                  HBELT
      IF  (IND.EQ.0)  NBSF = NBSF + 1                                   HBELT
      IF  (NPTPLY(JB).LE.0)  GO TO 31                                   HBELT
C                                                                       HBELT
C     FIRST LOOP ON K                                                   HBELT
C     COMPUTE Z(K),ZR(K),E3(K),U(K-1),BL(K-1),FB(K-1)                   HBELT
C     NEED     NL(K),BB(K-1)                                            HBELT
C     NOTE: AN INDEX K-1 REFERS TO BELT SEGMENT BETWEEN K-1 AND K.      HBELT
C                                                                       HBELT
      K1 = K2 + 1                                                       HBELT
      K2 = K2 + NPTPLY(JB)                                              HBELT
      DO  20  K=K1,K2                                                   HBELT
      KNL = KNLO + K                                                    HBELT
      KI = NL(1,KNL)                                                    HBELT
C                                                                       HBELT
C     HERE K   IS INDEX OF POINTS IN PLAY ON EACH HARNESS               HBELT
C          KNL IS INDEX OF ALL POINTS IN PLAY                           HBELT
C          KI  IS INDEX OF ALL POINTS                                   HBELT
C                                                                       HBELT
      KS   = IABS(IBAR(1,KI))                                           HBELT
      IF  (KS.GT.100)  NTP = 1                                          HBELT
      IF  (KS.GT.100)  KS = MOD(KS,100)                                 HBELT
```

```
      KE = IBAR(2,KI)                                                   HBELT
      CALL DOT31 (D(1,1,KS),BAR(4,KI),T1)                               HBELT
      CALL DOT31 (D(1,1,KS),BAR(7,KI),T2)                               HBELT
      DO  11  J=1,3                                                     HBELT
      R(J) = V(J)                                                       HBELT
      V(J) = BAR(J+3,KI) + BAR(J+6,KI)                                  HBELT
      TR(J,K) = T1(J)                                                   HBELT
      ZR(J,K) = T1(J) + T2(J)                                           HBELT
      S (J,2) = S(J,1)                                                  HBELT
   11 S (J,1) = SEGLP(J,KS) + ZR(J,K)                                   HBELT
      CALL CROSS (WMEG(1,KS),V,T)                                       HBELT
      IF  (KE.EQ.0)  GO TO 12                                           HBELT
      CALL MAT31 (BD(7,KE),BAR(4,KI),T2)                                HBELT
      CALL DOT31 (D(1,1,KS),T2,T1)                                      HBELT
   12 DO  13  J=1,3                                                     HBELT
      T(J) = T(J) + BAR(J+12,KI)                                        HBELT
   13 E(J,3,K) = T1(J)                                                  HBELT
      CALL DOT31 (D(1,1,KS),T,V)                                        HBELT
      DO  14  J=1,3                                                     HBELT
   14 V(J) = V(J) + SEGLV(J,KS)                                         HBELT
      FB(K) = 0.0                                                       HBELT
      FP(K) = 0.0                                                       HBELT
      IF  (K.EQ.K1)  GO TO 20                                           HBELT
      DO  15  J=1,3                                                     HBELT
   15 U(J,K-1) = S(J,1) - S(J,2)                                        HBELT
      BL(K-1) = DSQRT(U(1,K-1)**2 + U(2,K-1)**2 + U(3,K-1)**2)          HBELT
      DO  16  J=1,3                                                     HBELT
   16 U(J,K-1) = U(J,K-1)/BL(K-1)                                       HBELT
      STRAIN = (BL(K-1)/BB(KNL-1)) - 1.0                                HBELT
      IF  (STRAIN.LT.EPS(12)) STRAIN = 0.0                              MISDOT
      NT = NL(2,KNL)                                                    HBELT
      BLDOT = U(1,K-1)*(V(1)-R(1))                                      HBELT
     *       + U(2,K-1)*(V(2)-R(2))                                     HBELT
     *       + U(3,K-1)*(V(3)-R(3))                                     HBELT
      STRDOT = (BB(KNL-1)*BLDOT-BL(K-1)*BBDOT(KNL-1))/BB(KNL-1)**2      HBELT
      CALL FRCDFL (STRAIN,STRDOT,NT,0,FPK,ELOSS)                        HBELT
      CALL FRCDFL (STRAIN,STRDOT,NT,1,FBK,ELOSS)                        HBELT
      PTLOSS(1,K-1) = BB(KNL-1)*ELOSS                                   HBELT
      FP(K-1) = FPK                                                     HBELT
      FB(K-1) = FBK                                                     HBELT
      IF  (IND.NE.0)  GO TO 20                                          ENDPFX
      IF (K.NE.K1+1) GO TO 19                                           ENDPFX
      BSF(1,NBSF) = STRAIN                                              ENDPFX
      BSF(2,NBSF) = FBK                                                 ENDPFX
   19 IF (K.NE.K2) GO TO 20                                             ENDPFX
      BSF(3,NBSF) = STRAIN                                              ENDPFX
      BSF(4,NBSF) = FBK                                                 ENDPFX
   20 CONTINUE                                                          HBELT
C                                                                       HBELT
C     SECOND LOOP ON K                                                  HBELT
```

```
C     COMPUTE FCE(K),E1(K),E2(K),EDOT(K),FR(K),U1(KS),U2(KS)          HBELT
C     NEED    FB(K&K-1),U(K&K-1),ZR(K),E3(K)                           HBELT
C                                                                      HBELT
      DO 30  K=K1,K2                                                   HBELT
      KNL = KNL0 + K                                                   HBELT
      KI = NL(1,KNL)                                                   HBELT
      KS    = IABS(IBAR(1,KI))                                         HBELT
      IF  (KS.GT.100)  KS = MOD(KS,100)                                HBELT
      DO  21  J=1,3                                                    HBELT
      FCE(J,K) = 0.0                                                   BUTLER1
      IF (K.NE.K2) FCE(J,K) = FB(K)*U(J,K)                             BUTLER1
   21 IF  (K.NE.K1)  FCE(J,K) = FCE(J,K) - FB(K-1)*U(J,K-1)            HBELT
      NT = IBAR(3,KI)                                                  HBELT
      NF = NTAB(NT+5)                                                  HBELT
      IF  (NF.EQ.0 .AND. IND.EQ.0)  GO TO 30                           HBELT
      IF  (IBAR(4,KI).EQ.0)  GO TO 22                                  HBELT
      CALL DOT31 (D(1,1,KS),BAR(10,KI),T1)                             HBELT
      GO TO 24                                                         HBELT
   22 DO  23  J=1,3                                                    HBELT
      T1(J) = 0.0                                                      HBELT
      IF  (K.NE.K2)  T1(J) = U(J,K)                                    HBELT
   23 IF  (K.NE.K1)  T1(J) = T1(J) + U(J,K-1)                          HBELT
   24 CALL CROSS (T1,E(1,3,K),E(1,1,K))                                HBELT
      CALL CROSS (E(1,3,K),E(1,1,K),E(1,2,K))                          HBELT
      DO  25  J=1,3                                                    HBELT
      EDOT(J,K) = DSQRT(E(1,J,K)**2 + E(2,J,K)**2 + E(3,J,K)**2)       HBELT
      DO  25  I=1,3                                                    HBELT
   25 E(I,J,K) = E(I,J,K)/EDOT(J,K)                                    HBELT
      CALL DOT31 (E(1,1,K),FCE(1,K),FR(1,K))                           HBELT
   30 CONTINUE                                                         HBELT
   31 CONTINUE                                                         HBELT
      IF  (NTP.LE.0)  GO TO 41                                         HBELT
C                                                                      HBELT
C     SUM FCE,FR FOR TIE-POINTS                                        HBELT
C                                                                      HBELT
      KNL1 = KNL0 + 2                                                  HBELT
      KNL2 = KNL0 + K2                                                 HBELT
      DO  40  KNL=KNL1,KNL2                                            HBELT
      KI = NL(1,KNL)                                                   HBELT
      KS    = IABS(IBAR(1,KI))                                         HBELT
      IF  (KS.LT.100)  GO TO 40                                        HBELT
      KS1 = KS/100                                                     HBELT
      KH = KNL - KNL0                                                  HBELT
      MH = 0                                                           HBELT
      DO  38  JNL=KNL1,KNL                                             HBELT
      KI = NL(1,JNL-1)                                                 HBELT
      KS    = IABS(IBAR(1,KI))                                         HBELT
      IF  (KS.LT.100)  GO TO 38                                        HBELT
      KS2 = KS/100                                                     HBELT
      IF  (KS2.NE.KS1)  GO TO 38                                       HBELT
```

```
      JH = JNL-1 - KNLO                                                  HBELT
      IF  (MH.EQ.0)  MH = JH                                             HBELT
      DO  37  J=1,3                                                      HBELT
      IF  (MH.EQ.JH)  FCE(J,MH) = FCE(J,MH) + FCE(J,KH)                  HBELT
   37 FCE(J,JH) = FCE(J,MH)                                              HBELT
      CALL DOT31 (E(1,1,JH),FCE(1,JH),FR(1,JH))                          HBELT
   38 CONTINUE                                                           HBELT
      IF  (MH.EQ.0)  GO TO 40                                            HBELT
      KI = NL(1,KNL)                                                     HBELT
      IBAR(1,KI) = -IABS(IBAR(1,KI))                                     HBELT
      DO  39  J=1,3                                                      HBELT
   39 FCE(J,KH) = FCE(J,MH)                                              HBELT
      CALL DOT31 (E(1,1,KH),FCE(1,KH),FR(1,KH))                          HBELT
   40 CONTINUE                                                           HBELT
C                                                                        HBELT
C     IF CALL IS FROM SUBROUTINE CONTCT,                                 HBELT
C     ADD FORCES (FCE) MODIFIED BY FRICTION TO U1,U2 ARRAYS.             HBELT
C                                                                        HBELT
   41 IF  (IND.NE.0)  GO TO 52                                           HBELT
      K2 = 0                                                             HBELT
      DO  51  JB=J1,J2                                                   HBELT
      IF  (NPTPLY(JB).LE.0)  GO TO 51                                    HBELT
      K1 = K2 + 1                                                        HBELT
      K2 = K2 + NPTPLY(JB)                                               HBELT
      DO  50  K=K1,K2                                                    HBELT
      KNL = KNLO + K                                                     HBELT
      KI = NL(1,KNL)                                                     HBELT
      IF  (IBAR(1,KI).LT.0)  GO TO 50                                    HBELT
      KS = IBAR(1,KI)                                                    HBELT
      IF  (KS.GT.100)  KS = MOD(KS,100)                                  HBELT
      NT = IBAR(3,KI)                                                    HBELT
      NF = NTAB(NT+5)                                                    HBELT
      IF  (NF.EQ.0)  GO TO 43                                            HBELT
      DO  42  J=1,3                                                      HBELT
   42 T1(J) = FR(J,K)                                                    HBELT
      FR1 = TAB(NF+2)*DABS(T1(3))                                        HBELT
      FR2 = TAB(NF+4)*DABS(T1(3))                                        HBELT
      IF  (DABS(T1(1)).GT.FR1)  T1(1) = DSIGN(FR1,T1(1))                 HBELT
      IF  (DABS(T1(2)).GT.FR2)  T1(2) = DSIGN(FR2,T1(2))                 HBELT
      CALL MAT31 (E(1,1,K),T1,FCE(1,K))                                  HBELT
   43 CALL CROSS (ZR(1,K),FCE(1,K),T2)                                   HBELT
      CALL MAT31 (D(1,1,KS),T2,T1)                                       HBELT
      DO  44  J=1,3                                                      HBELT
      U1(J,KS) = U1(J,KS) + FCE(J,K)                                     HBELT
   44 U2(J,KS) = U2(J,KS) + T1(J)                                        HBELT
   50 CONTINUE                                                           HBELT
   51 CONTINUE                                                           HBELT
   52 KNLO = KNLO + K2                                                   HBELT
      CALL ELTIME (2,38)                                                 HBELT
      RETURN                                                             HBELT
      END                                                                HBELT
```

```
      SUBROUTINE HBPLAY                                                 HBPLAY
C                                                   REV III.5 10/17/85EDGE
      IMPLICIT REAL*8 (A-H,O-Z)                                         HBPLAY
      COMMON/CONTRL/ TIME,NSEG,NJNT,NPL,NBLT,NBAG,NVEH,NGRND,          HBPLAY
     *               NS,NQ,NSD,NFLX,NHRNSS,NWINDF,NJNTF,NPRT(36),NPG    PAGE
      COMMON/CNTSRF/ PL(24,30),BELT(20,8),TPTS(6,8),BD(24,40)           EDGE
      COMMON/SGMNTS/ D(3,3,30),WMEG(3,30),WMEGD(3,30),U1(3,30),U2(3,30),HBPLAY
     *               SEGLP(3,30),SEGLV(3,30),SEGLA(3,30),NSYM(30)       HBPLAY
      COMMON/HRNESS/ BAR(15,100),BB(100),BBDOT(100),PLOSS(2,100),       HBPLAY
     *               XLONG(20),HTIME(2),IBAR(5,100),NL(2,100),          HBPLAY
     *               NPTSPB(20),NPTPLY(20),NTHRNS(20),NBLTPH(5)         HBPLAY
C     THIS COMMON/TEMPVS/ IS SHARED BY HPTURB, HBPLAY, HBELT AND HSETC. HBPLAY
      COMMON/TEMPVS/ B(3,3,3),S(3,3),T(3),R(3),V(3),T1(3),T2(3),        HBPLAY
     *               E(3,3,50),EDOT(3,50),FCE(3,50),FR(3,50),ZR(3,50),  HBPLAY
     *               TR(3,50),U(3,50),PTLOSS(2,50),BL(50),FB(50),FP(50),HBPLAY
     *               OLDBB(100),RHS(3,54),C(3,3,200),IJK(54,54)         HBPLAY
      IF  (NHRNSS.LE.0)  GO TO 99                                       HBPLAY
C                                                                       HBPLAY
C     SAVE PREVIOUS NL,BB AND PLOSS ARRAYS.                             HBPLAY
C     USE IJK,OLDBB AND PTLOSS AS TEMP STORAGE.                         HBPLAY
C                                                                       HBPLAY
      DO  10  I=1,100                                                   HBPLAY
      IJK(I,1) = NL(1,I)                                                HBPLAY
      PTLOSS(I,1) = PLOSS(1,I)                                          HBPLAY
   10 OLDBB(I) = BB(I)                                                  HBPLAY
      JNL = 1                                                           HBPLAY
      J1 = 1                                                            HBPLAY
      K1 = 1                                                            HBPLAY
      LL = 0                                                            HBPLAY
      DO  90  NH=1,NHRNSS                                               HBPLAY
      IF  (NBLTPH(NH).LE.0)  GO TO 90                                   HBPLAY
      J2 = J1 + NBLTPH(NH) - 1                                          HBPLAY
      DO  80  NB=J1,J2                                                  HBPLAY
      L1 = LL                                                           HBPLAY
      IF  (NPTSPB(NB).LE.0)  GO TO 80                                   HBPLAY
      K2 = K1 + NPTSPB(NB) - 1                                          HBPLAY
      KB = 0                                                            HBPLAY
      DO  30  K=K1,K2                                                   HBPLAY
      KB = KB + 1                                                       HBPLAY
C                                                                       HBPLAY
C     HERE K IS INDEX OF ALL POINTS                                     HBPLAY
C         KB IS INDEX OF POINTS ON A SINGLE BELT                        HBPLAY
C         LL IS INDEX OF ALL POINTS IN PLAY                             HBPLAY
C         JB IS INDEX OF PREVIOUS POINT ON BELT IN PLAY                 HBPLAY
C                                                                       HBPLAY
      KS = IABS(IBAR(1,K))                                              HBPLAY
      IF  (KS.GT.100)  KS = MOD(KS,100)                                 HBPLAY
      CALL DOT31 (D(1,1,KS),BAR(4,K),T1)                                HBPLAY
      CALL DOT31 (D(1,1,KS),BAR(7,K),T2)                                HBPLAY
      DO  11  J=1,3                                                     HBPLAY
```

```
   11 U(J,KB) = SEGLP(J,KS) + T1(J) + T2(J)                               HBPLAY
      IF  (K.EQ.K1)  GO TO 30                                              HBPLAY
      LL = LL + 1                                                          HBPLAY
   12 JJ = NL(1,LL)                                                        HBPLAY
      JB = JJ - K1 + 1                                                     HBPLAY
      DSS = 0.0                                                            HBPLAY
      DO  13  J=1,3                                                        HBPLAY
      ZR(J,KB) = U(J,KB) - U(J,JB)                                         HBPLAY
   13 DSS = DSS + ZR(J,KB)**2                                              HBPLAY
      BL(LL) = DSQRT(DSS)                                                  HBPLAY
      IF  (JJ.EQ.K1 .OR. IABS(IBAR(1,JJ)).GT.100)  GO TO 30                HBPLAY
      JS = IBAR(1,JJ)                                                      HBPLAY
      JE = IBAR(2,JJ)                                                      HBPLAY
      IF  (JE.LE.0)  GO TO 30                                              HBPLAY
      CALL MAT31 (BD(7,JE),BAR(4,JJ),T2)                                   HBPLAY
      CALL DOT31 (D(1,1,JS),T2,R)                                          HBPLAY
      DPR = 0.0                                                            HBPLAY
      DO  17  J=1,3                                                        HBPLAY
   17 DPR = DPR + R(J)*(ZR(J,KB)/BL(LL) - ZR(J,JB)/BL(LL-1))               HBPLAY
      IF  (DPR.LT.0.0)  GO TO 30                                           HBPLAY
      LL = LL - 1                                                          HBPLAY
      GO TO 12                                                             HBPLAY
   30 NL(1,LL+1) = K                                                       HBPLAY
      L2 = L1 + 1                                                          HBPLAY
      LL = LL + 1                                                          HBPLAY
      L3 = LL-1                                                            HBPLAY
      DO  31  J=L2,LL                                                      HBPLAY
   31 NL(2,J) = NTHRNS(NB)                                                 HBPLAY
      IF  (XLONG(NB).EQ.0.0)  GO TO 35                                     HBPLAY
C                                                                          HBPLAY
C     FIRST TIME IN ROUTINE, SET INITIAL BB ARRAY.                         HBPLAY
C     INPUT XLONG MUST BE NON-ZERO TO TRIGGER THIS TEST.                   HBPLAY
C                                                                          HBPLAY
      XLG = 0.0                                                            HBPLAY
      DO  32  J=L2,L3                                                      HBPLAY
   32 XLG = XLG + BL(J)                                                    HBPLAY
      XLG = 1.0 + XLONG(NB)/XLG                                            HBPLAY
      DO  33  J=L2,L3                                                      HBPLAY
   33 BB(J) = XLG*BL(J)                                                    HBPLAY
      XLONG(NB) = 0.0                                                      HBPLAY
      GO TO 52                                                             HBPLAY
C                                                                          HBPLAY
C     DETERMINE IF NEW NL ARRAY IS DIFFERENT FROM PREVIOUS NL ARRAY.       HBPLAY
C     IF SO, RECOMPUTE BB ELEMENTS FOR POINTS THAT ARE DIFFERENT.          HBPLAY
C                                                                          HBPLAY
   35 IF  (NL(1,L2).EQ.IJK(JNL,1))  GO TO 61                               HBPLAY
      WRITE (6,62)                                                         HBPLAY
   62 FORMAT ('0 LOGIC ERROR IN SUB HBPLAY. PROGRAM TERMINATED.')          HBPLAY
      STOP 42                                                              HBPLAY
   61 LTEST = 0                                                            HBPLAY
```

```
      M = L2                                                            HBPLAY
      N = JNL                                                           HBPLAY
   36 IF  (NL(1,M+1)-IJK(N+1,1))  39,37,41                              HBPLAY
   37 BB(M) = OLDBB(N)                                                  HBPLAY
      PLOSS(1,M) = PTLOSS(N,1)                                          HBPLAY
   38 M = M+1                                                           HBPLAY
      N = N+1                                                           HBPLAY
      IF  (M-LL)  36,51,51                                              HBPLAY
C                                                                       HBPLAY
C     POINT M+1 IS NEW.                                                 HBPLAY
C                                                                       HBPLAY
   39 MO = M                                                            HBPLAY
      NO = N                                                            HBPLAY
      LTEST = 1                                                         HBPLAY
   40 M = M+1                                                           HBPLAY
C                                                                       CHGIII
C     MODIFY NEW POINT TO LIE IN BELT PLANE                             CHGIII
C                                                                       CHGIII
      IP1 = N - 1                                                       CHGIII
      IF  (N.GT.JNL)  GO TO 63                                          CHGIII
      IP1 = N                                                           CHGIII
C     (IS THIRD POINT AVAILABLE FROM OLD POINTS IN PLAY?)               CHGIII
      IF  (IJK(N+1,1).EQ.NL(1,LL))  GO TO 43                            CHGIII
   63 DO  64  I=1,3                                                     CHGIII
      IP = IP1 + I - 1                                                  CHGIII
C     (USE OLD POINTS IP = N-1,N,N+1 IF N > JNL                         CHGIII
C                  OR IP = N,N+1,N+2 IF N = JNL AND N+2 EXISTS)         CHGIII
      NI = IJK(IP,1)                                                    CHGIII
      NSI= IABS(IBAR(1,NI))                                             NSFIX
      IF (NSI.GT.100) NSI = MOD(NSI,100)                                NSFIX
      CALL DOT31 (D(1,1,NSI),BAR(4,NI),T1)                              NSFIX
      CALL DOT31 (D(1,1,NSI),BAR(7,NI),T2)                              NSFIX
      DO  64  J=1,3                                                     CHGIII
   64 S(J,I) = SEGLP(J,NSI)+ T1(J) + T2(J)                              NSFIX
      DO  65  J=1,3                                                     CHGIII
      S(J,3) = S(J,3) - S(J,2)                                          CHGIII
   65 S(J,2) = S(J,2) - S(J,1)                                          CHGIII
C     (S(*,1) IS POINT P1 IN INERTIAL REFERENCE)                        CHGIII
C     (S(*,2) IS VECTOR (P2-P1) IN INERTIAL REFERENCE)                  CHGIII
C     (S(*,3) IS VECTOR (P3-P2) IN INERTIAL REFERENCE)                  CHGIII
      CALL CROSS (S(1,3),S(1,2),T2)                                     CHGIII
      ABST = DSQRT(T2(1)**2 + T2(2)**2 + T2(3)**2)                      CHGIII
      DO  66  J=1,3                                                     CHGIII
   66 T2(J) = T2(J)/ABST                                                CHGIII
C     (T2 IS T, THE NORMALIZED PLANE VECTOR IN INERTIAL REFERENCE)      CHGIII
      MI = NL(1,M)                                                      CHGIII
      MS = IABS(IBAR(1,MI))                                             CHGIII
      IF (MS.GT.100)   MS = MOD(MS,100)                                 CHGIII
      ME = IBAR(2,MI)                                                   CHGIII
      CALL MAT31 (D(1,1,MS),T2,T1)                                      CHGIII
```

```
C     (T1 IS T IN ELLIPSOID REFERENCE OF NEW POINT M)                    CHGIII
      D1 = T2(1)*S(1,1) + T2(2)*S(2,1) + T2(3)*S(3,1)                    CHGIII
      D2 = T1(1)*BAR(7,MI) + T1(2)*BAR(8,MI) + T1(3)*BAR(9,MI)           CHGIII
      D3 = T2(1)*SEGLP(1,MS) + T2(2)*SEGLP(2,MS) + T2(3)*SEGLP(3,MS)     CHGIII
      DD = D1 - D2 - D3                                                  CHGIII
C     (DD IS D, THE DISTANCE OF ELLIPSOID CENTER TO PLANE)               CHGIII
      CALL MAT31 (BD(16,ME),T1,R)                                        CHGIII
      BX = DD/(T1(1)*R(1) + T1(2)*R(2) + T1(3)*R(3))                     CHGIII
      D4 = T1(1)*BAR(4,MI) + T1(2)*BAR(5,MI) + T1(3)*BAR(6,MI)           CHGIII
      DO  67  J=1,3                                                      CHGIII
      R(J) = BX*R(J)                                                     CHGIII
C     (R IS S, THE CENTER OF THE ELLIPSE)                                CHGIII
   67 V(J) = BAR(J+3,MI) + (DD-D4)*T1(J)                                 CHGIII
C     (BAR(J+3,MI) IS P, THE NEW POINT TO BE ADDED)                      CHGIII
C     (V IS Q, THE PROJECTION OF POINT P ONTO THE PLANE)                 CHGIII
      AX = DSQRT( (BX*DD-1.0) / (BX*DD-XDY(V,BD(7,ME),V)) )              CHGIII
      DO  68  J=1,3                                                      CHGIII
   68 BAR(J+3,MI) = R(J) + AX*(V(J)-R(J))                                CHGIII
C     (BAR(J+3,MI) IS R = S + A(Q - S), Q EXTENDED TO ELLIPSOID)         CHGIII
      GO TO 43                                                           HBPLAY
C                                                                        HBPLAY
C      POINT N+1 IS DROPPED.                                             HBPLAY
C                                                                        HBPLAY
   41 MO = M                                                             HBPLAY
      NO = N                                                             HBPLAY
      LTEST = 1                                                          HBPLAY
   42 N = N+1                                                            HBPLAY
   43 IF  (NL(1,M+1)-IJK(N+1,1))  40,44,42                               HBPLAY
C                                                                        HBPLAY
C     POINTS NO TO N+1 ARE BEING REPLACED WITH POINTS MO TO M+1.         HBPLAY
C                                                                        HBPLAY
   44 SUMBL = 0.0                                                        HBPLAY
      DO  45  J=MO,M                                                     HBPLAY
   45 SUMBL = SUMBL + BL(J)                                              HBPLAY
      SUMPL = 0.0                                                        HBPLAY
      SUMBB = 0.0                                                        HBPLAY
      DO  46  J=NO,N                                                     HBPLAY
      SUMPL = SUMPL + PTLOSS(J,1)                                        HBPLAY
   46 SUMBB = SUMBB + OLDBB(J)                                           HBPLAY
      RATPL = SUMPL/SUMBL                                                HBPLAY
      RATIO = SUMBB/SUMBL                                                HBPLAY
      DO  47  J=MO,M                                                     HBPLAY
      PLOSS(1,J) = RATPL*BL(J)                                           HBPLAY
   47 BB(J) = RATIO*BL(J)                                                HBPLAY
      GO TO 38                                                           HBPLAY
   51 JNL = N+1                                                          HBPLAY
      IF  (LTEST.EQ.0)  GO TO 79                                         HBPLAY
C                                                                        HBPLAY
C     PRINT NEW POINT ARRAY IF DIFFERENT.                                HBPLAY
C                                                                        HBPLAY
```

```
52   NPTS = LL - L1                                                    HBPLAY
     USEC = 1000.0*TIME                                                HBPLAY
     WRITE  (6,53)  USEC,NH,NB,NPTS,NTHRNS(NB)                         HBPLAY
53   FORMAT ('0 HBPLAY TIME =',F10.3,' MSEC. NH,NB,NPTS NT=',4I6)      HBPLAY
     WRITE  (6,54)  (NL(1,J),J=L2,LL)                                  HBPLAY
54   FORMAT ('   NL(1)=',15I8/(8X,15I8))                               HBPLAY
     WRITE  (6,55)  (BB(J),J=L2,L3)                                    HBPLAY
55   FORMAT ('   BB    =',6X,14F8.3/(6X,15F8.3))                       HBPLAY
79   K1 = K2 + 1                                                       HBPLAY
80   NPTPLY(NB) = LL - L1                                              HBPLAY
     J1 = J2 + 1                                                       HBPLAY
90   CONTINUE                                                          HBPLAY
99   RETURN                                                            HBPLAY
     END                                                               HBPLAY
```

```
      SUBROUTINE HEDING (LINES,LPP)                                     HEDING
C                                                  REV IV    02/01/88MISDOT
      IMPLICIT REAL*8 (A-H,O-Z)                                         HEDING
      COMMON/CONTRL/ TIME,NSEG,NJNT,NPL,NBLT,NBAG,NVEH,NGRND,           HEDING
     *               NS,NQ,NSD,NFLX,NHRNSS,NWINDF,NJNTF,NPRT(36),NPG    PAGE
      COMMON/JBARTZ/ MNPL(    30),MNBLT(    8),MNSEG(    30),MNBAG(     6), HEDING
     *               MPL(3,5,30),MBLT(3,5,8),MSEG(3,5,30),MBAG(3,10,6), HEDING
     *               NTPL( 5,30),NTBLT( 5,8),NTSEG( 5,30)               HEDING
      COMMON/TITLES/ DATE(3),COMENT(40),VPSTTL(20),BDYTTL(5),           HEDING
     *               BLTTTL(5,8),PLTTL(5,30),BAGTTL(5,6),SEG(30),       HEDING
     *               JOINT(30),CGS(30),JS(30)                           HEDING
      REAL DATE,COMENT,VPSTTL,BDYTTL,BLTTTL,PLTTL,BAGTTL,SEG,JOINT      HEDING
      LOGICAL*1 CGS,JS                                                  HEDING
      COMMON/FORCES/PSF(7,70),BSF(4,20),SSF(10,40),BAGSF(3,20),         NCFORC
     *               PRJNT(7,30),NPANEL(5),NPSF,NBSF,NSSF,NBGSF         HEDING
      COMMON/CNSNTS/ PI,RADIAN,G,THIRD,EPS(24),                         HEDING
     *               UNITL,UNITM,UNITT,GRAVTY(3),TWOPI                  TWOPI
      COMMON/RSAVE/  XSG(3,20,3),DPMI(3,3,30),LPMI(30),                 ATBIII
     *               NSG(9),MSG(20,9),MCG,MCGIN(24,5),KREF(20,9)        TTHKREF
      COMMON/DAMPER/ APSDM(3,20),APSDN(3,20),ASD(5,20),MSDM(20),MSDN(20)HEDING
      COMMON/HRNESS/ BAR(15,100),BB(100),BBDOT(100),PLOSS(2,100),       HEDING
     *               XLONG(20),HTIME(2),IBAR(5,100),NL(2,100),          HEDING
     *               NPTSPB(20),NPTPLY(20),NTHRNS(20),NBLTPH(5)         HEDING
C     NOTE: SUBROUTINES POSTPR & HEDING SHARE THIS COMMON/TEMPVS/.      HEDING
C     SEE COMMENT IN POSTPR ABOUT FIRST DIMENSION OF PLDATA.            HEDING
      REAL HEAD,PHED,BLANK,PLDATA,USEC,ZTTH,AHED,AHEAD,GHED,ZZZ         PLTINC
      COMMON/TEMPVS/ TDATA(14,65),HEAD(20),NOPL(150),MOPL(150),         CHGIII
     *     M1PL(150),PLDATA(97,20),USEC(45),ZZZ(1000,25),ZTTH(14,45,65) MISDOT
      LOGICAL   LOLD , LNEW                                             HEDING
      DIMENSION PHED(5),HEDJ(4,2),HEADJJ(4,2),HEADR(20)                 TTHKREF
      DATA HEDJ/8HIPIN  FL,8HEXURE  A,8HZIMUTH  ,8HTORSION ,            HEDING
     *          8HIEULER  ,8HPREC.  N,8HUTATION ,8H  SPIN  /            HEDING
      DIMENSION AHED(5,2),AHEAD(5,20),GHED(2)                           ACCEL
      DATA AHED/4H IN ,4H    ,4H REF,4HEREN,4HCE  ,                     ACCEL
     *          4H  AC,4HCELE,4HROME,4HTER ,4H    /                     ACCEL
      DATA GHED/4H(0G),4H(1G)/                                          ACCEL
      DATA BLANK/4H    /                                                HEDING
      DATA PHED/4HSPRF,4HPNL1,4HPNL2,4HPNL3,4HPNL4/                     HEDING
      NPRT4 = NPRT(4) + 4                                               HEDING
      IF  (NPRT4.LE.0 .OR. NPRT4.GT.8)  STOP 40                         HEDING
      GO TO  (11,11,82,12,12,11,11,12) , NPRT4                          HEDING
   11 LOLD = .FALSE.                                                    HEDING
      LNEW = .TRUE.                                                     HEDING
      GO TO 13                                                          HEDING
   12 LOLD = .TRUE.                                                     HEDING
      LNEW = .FALSE.                                                    HEDING
   13 MT = 20                                                           HEDING
      NLINES = MOD(LINES-1,LPP)+1                                       HEDING
      XPAGE = 0.01*FLOAT((LINES + LPP-1)/LPP)                           HEDING
C                                                                       HEDING
```

```
C     NOTE: MT WILL BE THE PAGE OR OUTPUT UNIT COUNTER                  HEDING
C           NT WILL BE THE ACTUAL OUTPUT UNIT NUMBER                    HEDING
C           IT WILL BE THE INDEX TO THE DATA ARRAY                      HEDING
C           NLINES WILL BE THE NUMBER OF LINES TO BE PRINTED            HEDING
C                                                                       HEDING
C                                                                       HEDING
C     EVERY LPP LINES PRINT HEADINGS FOR 9 TYPES OF OUTPUT ABOVE.       WINDOP
C                                                                       HEDING
      DO 20 K=1,9                                                       WINDOP
      IF (NSG(K).LE.0) GO TO 20                                         HEDING
      KSG = NSG(K)                                                      HEDING
      IF (K.EQ.9) GO TO 455                                             WINDOP
      J3  = 3                                                           HEDING
      IF (K.EQ.7) J3 = 2                                                HEDING
      DO 19 J1=1,KSG,J3                                                 HEDING
      MT = MT + 1                                                       HEDING
      NT = MT                                                           HEDING
      IF  (LNEW)  NT = 6                                                HEDING
      IT = MT - 20                                                      HEDING
      PAGE = FLOAT(MT) + XPAGE                                          HEDING
C     P & E PRINTER CARRIAGE CONTROL                                    PECONV
      CALL CARCON(NT,1)                                                 PECONV
      IF (NT.EQ.6) WRITE(NT,121) DATE,BLANK,NPG                         PAGE
      IF (NT.NE.6) WRITE(NT,121) DATE                                   PAGE
      IF (NT.EQ.6) NPG=NPG+1                                            PAGE
      WRITE (NT,21) COMENT,PAGE,VPSTTL,BDYTTL                           PAGE
      IF (K.EQ.1) WRITE (NT,22)                                         TTHKREF
      IF (K.EQ.2) WRITE (NT,23) UNITL,UNITT                             TTHKREF
      IF (K.EQ.3) WRITE (NT,24) UNITL                                   TTHKREF
      IF (K.EQ.4) WRITE (NT,25) UNITT                                   TTHKREF
      IF (K.EQ.5) WRITE (NT,26) UNITT                                   TTHKREF
      IF (K.EQ.6) WRITE (NT,27)                                         TTHKREF
      IF (K.EQ.7) WRITE (NT,28)                                         HEDING
      IF (K.EQ.8) WRITE(NT,200) UNITM                                   TTHKREF
      J2 = MIN0(J1+J3-1,KSG)                                            HEDING
      DO 14 J=J1,J2                                                     HEDING
      KK = MSG(J,K)                                                     HEDING
      HEAD(J) = SEG(IABS(KK))                                           ACCEL
      IF ((K.LT.7).OR.(K.EQ.8)) GO TO 214                               TTHKREF
      KK = IABS(KK)                                                     HEDING
      HEAD(J) = JOINT(KK)                                               HEDING
      JJ2 = J-J1+1                                                      HEDING
      K2 = 1                                                            HEDING
      IF (MSG(J,K).LT.0) K2 = 2                                         HEDING
      DO  35  K1=1,4                                                    HEDING
   35 HEADJJ(K1,JJ2) = HEDJ(K1,K2)                                      HEDING
      GO TO 14                                                          TTHKREF
  214 IF (MSG(J,K).LT.0) GOTO 302                                       ACCEL
      IF (KREF(J,K).EQ.0) HEADR(J)=SEG(NVEH)                            ACCEL
      IF (K.EQ.8) HEADR(J)=SEG(NGRND)                                   TTHKREF
```

```
      IF (K.EQ.1 .OR. K.EQ.4) HEADR(J)=SEG(KK)                          TTHKREF
      IF (KREF(J,K).NE.0) HEADR(J)=SEG(KREF(J,K))                       TTHKREF
      DO 301 II=1,5                                                     ACCEL
  301 AHEAD(II,J)=AHED(II,1)                                            ACCEL
      AHEAD(2,J)=HEADR(J)                                               ACCEL
      GOTO 14                                                           ACCEL
  302 HEADR(J)=SEG(IABS(KK))                                            ACCEL
      DO 303 II=1,4                                                     ACCEL
  303 AHEAD(II,J)=AHED(II,2)                                            ACCEL
      AHEAD(5,J)=GHED(KREF(J,K)+1)                                      ACCEL
   14 CONTINUE                                                          HEDING
      IF (K.LE.3) WRITE (NT,29) (BLANK,(XSG(I,J,K),I=1,3),J=J1,J2)      HEDING
      IF (K.LE.6) WRITE (NT,30) (BLANK,MSG(J,K),HEAD(J),J=J1,J2)        HEDING
      IF (K.EQ.8) WRITE (NT,30) (BLANK,MSG(J,K),HEAD(J),J=J1,J2)        WINDOP
      IF (K.LE.6 .OR. K.EQ.8) WRITE (NT,230)                            ACCEL
     *                        (BLANK,(AHEAD(II,J),II=1,5),J=J1,J2)      ACCEL
      IF ((K.LE.5).OR.(K.EQ.8)) WRITE (NT,31) (BLANK,J=J1,J2)           WINDOP
      IF (K.EQ.6) WRITE (NT,32) (BLANK,J=J1,J2)                         HEDING
      IF ((K.LT.7).OR.(K.EQ.8)) GOTO 15                                 WINDOP
      WRITE (NT,33) (BLANK,MSG(J,K),HEAD(J),J=J1,J2)                    HEDING
      WRITE (NT,36) (BLANK,UNITL,UNITM,J=J1,J2)                         HEDING
      WRITE (NT,37) (BLANK,(HEADJJ(K1,J),K1=1,4),J=1,JJ2)               HEDING
   15 WRITE (NT,38)                                                     HEDING
      IF  (.NOT.LNEW)  GO TO 19                                         HEDING
      IF  (K.EQ.7)  GO TO 17                                            HEDING
      JJ = 4*(J2-J1+1)                                                  HEDING
      DO  16  I=1,NLINES                                                HEDING
   16 WRITE (NT,39)  USEC(I),(ZTTH(J,I,IT),J=1,JJ)                      HEDING
      GO TO 19                                                          HEDING
   17 JJ = 7*(J2-J1+1)                                                  HEDING
      DO  18  I=1,NLINES                                                HEDING
   18 WRITE (NT,40)  USEC(I),(ZTTH(J,I,IT),J=1,JJ)                      HEDING
   19 CONTINUE                                                          HEDING
      GO TO 20                                                          CHGIII
C                                                                       CHGIII
C     PRINT HEADING FOR JOINT FORCES & TORQUES                          CHGIII
C                                                                       CHGIII
  455 CONTINUE                                                          CHGIII
      DO 860 II=1,KSG                                                   CHGIII
      IF(KREF(II,K).EQ.0) KRF = NVEH                                    TTHKREF
      IF(KREF(II,K).NE.0) KRF = KREF(II,K)                              TTHKREF
      JRF = MSG(II,9)                                                   WINDOP
      MT = MT + 1                                                       CHGIII
      NT = MT                                                           CHGIII
      IF (LNEW) NT = 6                                                  CHGIII
C     P & E CARRIAGE CONTROL                                            PECONV
      CALL CARCON(NT,1)                                                 PECONV
      IT = MT - 20                                                      CHGIII
      PAGE = FLOAT (MT) + XPAGE                                         CHGIII
      IF (NT.EQ.6) WRITE(NT,121) DATE,BLANK,NPG                         PAGE
```

```
      IF (NT.NE.6) WRITE(NT,121) DATE                                  PAGE
      IF (NT.EQ.6) NPG=NPG+1                                           PAGE
      WRITE (NT,21) COMENT,PAGE,VPSTTL,BDYTTL                          PAGE
      WRITE (NT,850) JOINT(JRF),SEG(JRF+1),SEG(KRF)                    OUT385
      WRITE (NT,38)                                                    CHGIII
      WRITE (NT,851) UNITM,UNITL,UNITM                                 CHGIII
      WRITE (NT,852)                                                   CHGIII
      WRITE (NT,38)                                                    CHGIII
      IF (.NOT.LNEW) GO TO 857                                         CHGIII
      DO 858  JK=1,NLINES                                              CHGIII
      WRITE (NT,856) USEC(JK),(ZTTH(J,JK,IT),J=1,6)                    CHGIII
  858 CONTINUE                                                         CHGIII
  857 CONTINUE                                                         CHGIII
  850 FORMAT(' '/47X,                                                  TTHKREF
     * A4,' JOINT FORCES & TORQUES ON ',A4,' IN ',A4,' REFERENCE')     OUT385
  851 FORMAT(4X,4HTIME,7X,13HJOINT FORCE (,A4,7H 10**2),10X,           CHGIII
     *14HJOINT TORQUE (,A4,1H-,A4,7H 10**2))                           CHGIII
  852 FORMAT(3X,6H(MSEC),8X,1HX,8X,1HY,8X,1HZ,14X,1HX,11X,1HY,11X,1HZ) CHGIII
  856 FORMAT(F9.3,3X,3F9.3,3X,3(2X,D10.3))                             CHGIII
  860 CONTINUE                                                         CHGIII
   20 CONTINUE                                                         HEDING
  121 FORMAT('1',18X,'DATE:',3X,4A4,80X,'PAGE',I5)                     PAGE
   21 FORMAT(8X,'RUN DESCRIPTION:',3X,20A4/27X,20A4,'PAGE:',F6.2/      PAGE
     *       3X,'VEHICLE DECELERATION:',3X,20A4/                       HEDING
     *       11X,'CRASH VICTIM:',3X,5A4 )                              HEDING
   22 FORMAT(' '47X,                                                   TTHKREF
     * 'POINT TOTAL ACCELERATION (G''S)'/)                             TTHKREF
   23 FORMAT(' '47X,                                                   TTHKREF
     *'POINT REL. VELOCITY (',A4,'/',A4,')'/)                          TTHKREF
   24 FORMAT(' '47X,                                                   TTHKREF
     *'POINT REL. LINEAR DISPLACEMENT (',A4,')'/)                      TTHKREF
   25 FORMAT(' '/47X,                                                  TTHKREF
     *'SEGMENT ANGULAR ACCELERATION (REV/',A4,'**2)'/)                 TTHKREF
   26 FORMAT(' '/47X,                                                  TTHKREF
     *'SEGMENT REL. ANGULAR VELOCITY (REV/',A4,')'/)                   TTHKREF
   27 FORMAT(' '/47X,                                                  TTHKREF
     * 'SEGMENT REL. ANGULAR DISPLACEMENT (DEG)'/)                     TTHKREF
   28 FORMAT(' '/47X,'JOINT PARAMETERS'/)                              TTHKREF
  200 FORMAT(' '/47X,'SEGMENT WIND FORCE (',A4,')'/)                   TTHKREF
   29 FORMAT(9X,3(A4,3X,'POINT (',F6.2,',',F6.2,',',F6.2,') ON ') )    HEDING
   30 FORMAT('           ',3(A4,9X,'SEGMENT NO.',I3,' - ',A4,5X)   )   TTHKREF
  230 FORMAT('    TIME ',3(A4,9X,5A4,6X))                              ACCEL
   31 FORMAT('   (MSEC)',3(A4,5X,'X',8X,'Y',8X,'Z',7X,'RES',1X)  )     HEDING
   32 FORMAT('   (MSEC)',3(A4,4X,'YAW',5X,'PITCH',5X,'ROLL',5X,'RES ') )HEDING
   33 FORMAT(9X,2(A1,21X,'JOINT NO.',I3,' - ',A4,20X)  )               HEDING
   36 FORMAT('    TIME ',2(A1,'STATE',5X,'JOINT ANGLES (DEG)',8X,      HEDING
     *                     'TOTAL TORQUE (',2A4,') ') )                HEDING
   37 FORMAT('   (MSEC)',2(A1,4A8,4X,'SPRING  VISCOUS    RES. ') )      HEDING
   38 FORMAT(1X)                                                       HEDING
   39 FORMAT(F9.3,3(3X,4F9.3) )                                        HEDING
```

```
   40 FORMAT(F9.3,2(F5.0,3F9.3,2X,3F9.3))                                HEDING
C                                                                      ATBIII
C     PRINT BODY PROPERTIES CONTROLLED BY H.10 CARDS                   WINDOP
C                                                                      ATBIII
      IF (MCG.EQ.0) GO TO 131                                          ATBIII
      DO 130 NCG=1,MCG                                                 ATBIII
      MT = MT +1                                                       ATBIII
      NT = MT                                                          ATBIII
      IF (LNEW) NT = 6                                                 ATBIII
C     P & E CARRIAGE CONTROL                                           PECONV
      CALL CARCON(NT,1)                                                PECONV
      IT = MT - 20                                                     ATBIII
      PAGE = FLOAT(MT) + XPAGE                                         ATBIII
      IF (NT.EQ.6) WRITE(NT,121) DATE,BLANK,NPG                        PAGE
      IF (NT.NE.6) WRITE(NT,121) DATE                                  PAGE
      IF (NT.EQ.6) NPG=NPG+1                                           PAGE
      WRITE (NT,21) COMENT,PAGE,VPSTTL,BDYTTL                          PAGE
      M = MCGIN(1,NCG)                                                 ATBIII
      WRITE (NT,132) M,SEG(M)                                          ATBIII
      N = MCGIN(2,NCG)                                                 ATBIII
      WRITE (NT,133) (MCGIN(I+2,NCG),I=1,N)                            ATBIII
      WRITE (NT,38)                                                    ATBIII
      WRITE (NT,134) UNITL,UNITM,UNITT,UNITL,UNITM,UNITT,UNITM,UNITL   KINETIC
      WRITE (NT,38)                                                    ATBIII
      IF (.NOT.LNEW) GO TO 130                                         ATBIII
      DO 129 I=1,NLINES                                                ATBIII
  129 WRITE (NT,135) USEC(I),(ZTTH(J,I,IT),J=1,12)                     KINETIC
  130 CONTINUE                                                         ATBIII
  131 CONTINUE                                                         ATBIII
  132 FORMAT(' ',47X,39HBODY PROPERTIES - REFERENCE SEGMENT NO.,       TTHKREF
     *        I3,2H (,A4,1H) )                                         ATBIII
  133 FORMAT(15X,21HINCLUDED SEGMENT NOS:,20I3)                        ATBIII
  134 FORMAT(14X,17HCENTER OF GRAVITY,13X,15HLINEAR MOMENTUM,17X,      KINETIC
     *        16HANGULAR MOMENTUM,18X,14HKINETIC ENERGY/               KINETIC
     *        4X,4HTIME,11X,1H(,A4,1H),21X,1H(,A4,1H-,A4,1H),19X,      KINETIC
     *        1H(,A4,1H-,A4,1H-,A4,1H),20X,1H(,A4,1H-,A4,1H)/          MISC
     *        3X,6H(MSEC),5X,1HX,7X,1HY,7X,1HZ,                        KINETIC
     *        2(10X,1HX,10X,1HY,10X,1HZ),6X,6HLINEAR,5X,               KINETIC
     *        7HANGULAR,5X,5HTOTAL)                                    KINETIC
  135 FORMAT(F9.3,3F8.3,9(1X,D10.3))                                   KINETIC
C                                                                      HEDING
C     PLANE FORCES HEADINGS                                            HEDING
C                                                                      HEDING
      MPSF = 0                                                         HEDING
      IF (NPL.EQ.0) GO TO 52                                           HEDING
      IF (NPRT(18).EQ.1.OR.NPRT(18).EQ.7) GO TO 52                     VARTTH
      IF (NPRT(18).EQ.10.OR.NPRT(18).EQ.11) GO TO 52                   VARTTH
      IF (NPRT(18).GE.14) GO TO 52                                     VARTTH
      DO  42  J=1,NPL                                                  HEDING
      IF (MNPL(J).EQ.0) GO TO 42                                       HEDING
```

```
      KPL = MNPL(J)                                                     HEDING
      DO 41 I=1,KPL                                                     HEDING
      MPSF = MPSF+1                                                     HEDING
      NOPL(MPSF) = J                                                    HEDING
      IF (MPL(3,I,J).LT.0) M1PL(MPSF) = MPL(2,I,J)                      CHGIII
      IF (MPL(3,I,J).GE.0) M1PL(MPSF) = MPL(1,I,J)                      CHGIII
   41 MOPL(MPSF) = MPL(2,I,J)                                           HEDING
   42 CONTINUE                                                          HEDING
      IF (MPSF.EQ.0) GO TO 52                                           HEDING
      DO 44 J1=1,MPSF,2                                                 HEDING
      J2 = MIN0(J1+1,MPSF)                                              HEDING
      MT = MT + 1                                                       HEDING
      NT = MT                                                           HEDING
      IF  (LNEW)  NT = 6                                                HEDING
C     P & E CARRIAGE CONTROL                                            PECONV
      CALL CARCON(NT,1)                                                 PECONV
      IT = MT - 20                                                      HEDING
      PAGE = FLOAT(MT) + XPAGE                                          HEDING
      IF (NT.EQ.6) WRITE(NT,121) DATE,BLANK,NPG                         PAGE
      IF (NT.NE.6) WRITE(NT,121) DATE                                   PAGE
      IF (NT.EQ.6) NPG=NPG+1                                            PAGE
      WRITE (NT,21) COMENT,PAGE,VPSTTL,BDYTTL                           PAGE
      WRITE (NT,45)                                                     HEDING
      N1 = NOPL(J1)                                                     HEDING
      N2 = NOPL(J2)                                                     HEDING
      M1 = MOPL(J1)                                                     HEDING
      M2 = MOPL(J2)                                                     HEDING
      MM1 = M1PL(J1)                                                    CHGIII
      MM2 = M1PL(J2)                                                    CHGIII
      IF  (J1.EQ.J2)  WRITE  (NT,46)                                    HEDING
     *               BLANK,N1,( PLTTL(I,N1),I=1,5),M1,SEG(M1)           HEDING
      IF  (J1.NE.J2)  WRITE  (NT,46)                                    HEDING
     *               BLANK,N1,( PLTTL(I,N1),I=1,5),M1,SEG(M1),          HEDING
     *               BLANK,N2,( PLTTL(I,N2),I=1,5),M2,SEG(M2)           HEDING
      WRITE (NT,47) (BLANK,UNITL,J=J1,J2)                               HEDING
      IF (J1.EQ.J2) WRITE (NT,48) BLANK,SEG(MM1)                        CHGIII
      IF (J1.NE.J2) WRITE (NT,448) BLANK,SEG(MM1),BLANK,SEG(MM2)        CHGIII
      WRITE (NT,49) (BLANK,UNITL,UNITM,UNITM,UNITM,J=J1,J2)             HEDING
      WRITE (NT,38)                                                     HEDING
      IF  (.NOT.LNEW)  GO TO 44                                         HEDING
      JJ = 7*(J2-J1+1)                                                  HEDING
      DO  43  I=1,NLINES                                                HEDING
   43 WRITE (NT,50)  USEC(I),(ZTTH(J,I,IT),J=1,JJ)                      HEDING
   44 CONTINUE                                                          HEDING
   45 FORMAT(27X,'CONTACT FORCES - SEGMENT PANELS VS. SEGMENTS' )       CHGIII
   46 FORMAT(' '/8X,2(A4,'  PANEL',I3,' (',5A4,') VS. SEGMENT',I3,      HEDING
     *        ' (',A4,') ') )                                           HEDING
   47 FORMAT(' ',8X,A4,'DEFL-  NORMAL  FRICTION RESULTANT  CONTACT LOCATHEDING
     *ION (',A4,')',A2,'DEFL-  NORMAL  FRICTION RESULTANT  CONTACT LOCATHEDING
     *ION (',A4,')')                                                    HEDING
```

```
   48 FORMAT('    TIME',A4,'ECTION   FORCE    FORCE    FORCE      (',A4 CHGIII
     *,' REFERENCE)'  )                                             CHGIII
  448 FORMAT('    TIME',A4,'ECTION   FORCE    FORCE    FORCE      (',A4 CHGIII
     *,' REFERENCE)',2X,A4,'ECTION   FORCE    FORCE    FORCE      (',A4 CHGIII
     *,' REFERENCE)'  )                                             CHGIII
   49 FORMAT('    (MSEC)',2(A3,'(',A4,')',2X,'(',A4,')',4X,'(',A4,')',3X,HEDING
     *        '(',A4,')     X       Y       Z  ') )                 HEDING
   50 FORMAT(F9.3,2(F9.3,3F9.2,3F8.3) )                             HEDING
   51 FORMAT(3X,'(MSEC)',4(A1,9X,'X',8X,'Y',8X,'Z',1X))             HEDING
C                                                                   HEDING
C     BELT FORCES HEADINGS                                          HEDING
C                                                                   HEDING
   52 MBSF = 0                                                      HEDING
      IF (NPRT(18).EQ.2.OR.NPRT(18).GE.13) GO TO 83                 VARTTH
      IF (NPRT(18).GE.7.AND.NPRT(18).LE.9) GO TO 83                 VARTTH
      IF  (NBLT.EQ.0)  GO TO 83                                     HEDING
      DO  54  J=1,NBLT                                              HEDING
      IF (MNBLT(J).EQ.0) GO TO 54                                   HEDING
      MBSF = MBSF+1                                                 HEDING
      NOPL(MBSF) = J                                                HEDING
      MOPL(MBSF) = MBLT(2,1,J)                                      HEDING
   54 CONTINUE                                                      HEDING
      IF  (MBSF.EQ.0)  GO TO 83                                     HEDING
      DO 56 J1=1,MBSF,2                                             HEDING
      J2 = MINO(J1+1,MBSF)                                          HEDING
      MT = MT + 1                                                   HEDING
      NT = MT                                                       HEDING
      IF  (LNEW)  NT = 6                                            HEDING
C     P & E CARRIAGE CONTROL                                        PECONV
      CALL CARCON(NT,1)                                             PECONV
      IT = MT - 20                                                  HEDING
      PAGE = FLOAT(MT) + XPAGE                                      HEDING
      IF (NT.EQ.6) WRITE(NT,121) DATE,BLANK,NPG                     PAGE
      IF (NT.NE.6) WRITE(NT,121) DATE                               PAGE
      IF (NT.EQ.6) NPG=NPG+1                                        PAGE
      WRITE (NT,21) COMENT,PAGE,VPSTTL,BDYTTL                       PAGE
      WRITE (NT,57)                                                 HEDING
      N1 = NOPL(J1)                                                 HEDING
      N2 = NOPL(J2)                                                 HEDING
      M1 = MOPL(J1)                                                 HEDING
      M2 = MOPL(J2)                                                 HEDING
      IF  (J1.EQ.J2)  WRITE  (NT,58)                                HEDING
     *               BLANK,N1,(BLTTTL(I,N1),I=1,5),M1,SEG(M1)       HEDING
      IF  (J1.NE.J2)  WRITE  (NT,58)                                HEDING
     *               BLANK,N1,(BLTTTL(I,N1),I=1,5),M1,SEG(M1),      HEDING
     *               BLANK,N2,(BLTTTL(I,N2),I=1,5),M2,SEG(M2)       HEDING
      WRITE (NT,59) (BLANK,J=J1,J2)                                 HEDING
      WRITE (NT,60) (BLANK,J=J1,J2)                                 HEDING
      WRITE (NT,61) (BLANK,UNITL,UNITL,UNITM,UNITL,UNITL,UNITM,J=J1,J2) HEDING
      WRITE (NT,38)                                                 HEDING
```

```
      IF  (.NOT.LNEW)  GO TO 56                                          HEDING
      JJ = 4*(J2-J1+1)                                                   HEDING
      DO  55  I=1,NLINES                                                 HEDING
   55 WRITE (NT,62)  USEC(I),(ZTTH(J,I,IT),J=1,JJ)                       HEDING
   56 CONTINUE                                                           HEDING
   57 FORMAT('0',26X,'CONTACT FORCES - BELTS VS. SEGMENTS')              HEDING
   58 FORMAT(' ',7X,2(A4,'    BELT',I3,' (',5A4,') VS. SEGMENT',I3,      HEDING
     *            ' (',A4,') ') )                                        HEDING
   59 FORMAT(' ',2X,2(A4,11X,'ANCHOR  POINT  A',14X,'ANCHOR  POINT  B'))HEDING
   60 FORMAT(4X,'TIME',2(A4,5X,'STRAIN',7X,'FORCE',12X,                  HEDING
     *                         'STRAIN',7X,'FORCE', 3X) )                HEDING
   61 FORMAT(3X,'(MSEC)',2(A4,2X,'(',A4,'/',A4,')',4X,'(',A4,')',9X,     HEDING
     *                         '(',A4,'/',A4,')',4X,'(',A4,')',3X) )     HEDING
   62 FORMAT(F9.3,4(F15.6,F12.2,3X) )                                    HEDING
C                                                                        HEDING
C     HARNESS BELT ENDPOINTS FORCES HEADINGS                             HEDING
C                                                                        HEDING
   83 IF  (NHRNSS.LE.0)  GO TO 91                                        HEDING
      MBSF = 0                                                           HEDING
      IF (NPRT(18).EQ.3.OR.NPRT(18).EQ.11) GO TO 91                      VARTTH
      IF (NPRT(18).EQ.9.OR.NPRT(18).EQ.8) GO TO 91                       VARTTH
      IF (NPRT(18).EQ.13.OR.NPRT(18).EQ.14) GO TO 91                     VARTTH
      IF (NPRT(18).GE.16) GO TO 91                                       VARTTH
      J1 = 1                                                             HEDING
      K1 = 1                                                             HEDING
      DO  85  I=1,NHRNSS                                                 HEDING
      IF  (NBLTPH(I).LE.0)  GO TO 85                                     HEDING
      J2 = J1 + NBLTPH(I) - 1                                            HEDING
      DO  84  J=J1,J2                                                    HEDING
      MBSF = MBSF + 1                                                    HEDING
      IF  (NPTSPB(J).LE.0)  GO TO 84                                     HEDING
      K2 = K1 + NPTSPB(J) - 1                                            HEDING
      NOPL(2*MBSF-1) = J                                                 HEDING
      NOPL(2*MBSF  ) = I                                                 HEDING
      MOPL(2*MBSF-1) = K1                                                HEDING
      MOPL(2*MBSF  ) = K2                                                HEDING
      K1 = K2 + 1                                                        HEDING
   84 CONTINUE                                                           HEDING
      J1 = J2 + 1                                                        HEDING
   85 CONTINUE                                                           HEDING
      DO  87  J1=1,MBSF,2                                                HEDING
      J2 = MIN0(J1+1,MBSF)                                               HEDING
      MT = MT + 1                                                        HEDING
      NT = MT                                                            HEDING
      IF  (LNEW)  NT = 6                                                 HEDING
C     P & E CARRIAGE CONTROL                                             PECONV
      CALL CARCON(NT,1)                                                  PECONV
      IT = MT - 20                                                       HEDING
      PAGE = FLOAT(MT) + XPAGE                                           HEDING
      IF (NT.EQ.6) WRITE(NT,121) DATE,BLANK,NPG                          PAGE
```

```
      IF (NT.NE.6) WRITE(NT,121) DATE                                 PAGE
      IF (NT.EQ.6) NPG=NPG+1                                          PAGE
      WRITE (NT,21) COMENT,PAGE,VPSTTL,BDYTTL                         PAGE
      WRITE (NT,88)                                                   HEDING
      WRITE (NT,89) (BLANK,NOPL(2*J-1),NOPL(2*J),J=J1,J2)             HEDING
      WRITE (NT,90) (BLANK,MOPL(2*J-1),MOPL(2*J),J=J1,J2)             HEDING
      WRITE (NT,60) (BLANK,J=J1,J2)                                   HEDING
      WRITE (NT,61) (BLANK,UNITL,UNITL,UNITM,UNITL,UNITL,UNITM,J=J1,J2) HEDING
      WRITE (NT,38)                                                   HEDING
      IF  (.NOT.LNEW)  GO TO 87                                       HEDING
      JJ = 4*(J2-J1+1)                                                HEDING
      DO  86  I=1,NLINES                                              HEDING
   86 WRITE (NT,62)  USEC(I),(ZTTH(J,I,IT),J=1,JJ)                    HEDING
   87 CONTINUE                                                        HEDING
   88 FORMAT('0',26X,'HARNESS SYSTEM BELT ENDPOINT FORCES')           HEDING
   89 FORMAT(9X,2(A4,11X,'BELT NO.',I4,' OF HARNESS NO.',I3,15X))     HEDING
   90 FORMAT(9X,2(A4,6X,'POINT NO.',I5,16X,'POINT NO.',I5,6X))        HEDING
C                                                                     HEDING
C     SPRING DAMPER FORCES HEADINGS                                   HEDING
C                                                                     HEDING
   91 IF  (NSD.LE.0)  GO TO 63                                        HEDING
      IF (NPRT(18).EQ.4.OR.NPRT(18).EQ.9) GO TO 63                    VARTTH
      IF (NPRT(18).GE.12) GO TO 63                                    VARTTH
      DO  94  J1=1,NSD,4                                              HEDING
      J2 = MIN0(J1+3,NSD)                                             HEDING
      MT = MT + 1                                                     HEDING
      NT = MT                                                         HEDING
      IF  (LNEW)  NT = 6                                              HEDING
C     P & E CARRIAGE CONTROL                                          PECONV
      CALL CARCON(NT,1)                                               PECONV
      IT = MT - 20                                                    HEDING
      PAGE = FLOAT(MT) + XPAGE                                        HEDING
      IF (NT.EQ.6) WRITE(NT,121) DATE,BLANK,NPG                       PAGE
      IF (NT.NE.6) WRITE(NT,121) DATE                                 PAGE
      IF (NT.EQ.6) NPG=NPG+1                                          PAGE
      WRITE (NT,21) COMENT,PAGE,VPSTTL,BDYTTL                         PAGE
      WRITE (NT,95) (BLANK,J,J=J1,J2)                                 HEDING
      DO  92  J=J1,J2                                                 HEDING
      M1 = MSDM(J)                                                    HEDING
      N1 = MSDN(J)                                                    HEDING
C     POSSIBLE OVERFLOW INTO NOPL ARRAY IS INTENTIONAL.               HEDING
      HEAD(2*J-1) = SEG(M1)                                           HEDING
   92 HEAD(2*J  ) = SEG(N1)                                           HEDING
      WRITE (NT,96)(BLANK,MSDM(J),HEAD(2*J-1),MSDN(J),HEAD(2*J),J=J1,J2)HEDING
      WRITE (NT,97) (BLANK,J=J1,J2)                                   HEDING
      WRITE (NT,98) (BLANK,UNITL,UNITM,J=J1,J2)                       HEDING
      WRITE (NT,38)                                                   HEDING
      IF  (.NOT.LNEW)  GO TO 94                                       HEDING
      JJ = 2*(J2-J1+1)                                                HEDING
      DO  93  I=1,NLINES                                              HEDING
```

```
   93 WRITE (NT,99)  USEC(I),(ZTTH(J,I,IT),J=1,JJ)                    HEDING
   94 CONTINUE                                                         HEDING
   95 FORMAT('0',26X,'SPRING DAMPER FORCES'/                           HEDING
     *        9X,4(A3,3X,'SPRING DAMPER NO.',I3,4X))                   HEDING
   96 FORMAT(9X,4(A3,'SEG',I3,'(',A4,') - SEG',I3,'(',A4,')'))         HEDING
   97 FORMAT(4X,'TIME',1X,4(A3,5X,'LENGTH',7X,'FORCE',4X))             HEDING
   98 FORMAT(3X,'(MSEC)',4(A3,5X,'(',A4,')',6X,'(',A4,')',4X))         HEDING
   99 FORMAT (F9.3,4(F14.3,F12.2,4X))                                  HEDING
C                                                                      HEDING
C     SEGMENT FORCES HEADINGS                                          HEDING
C                                                                      HEDING
   63 MSSF = 0                                                         HEDING
      IF (NPRT(18).EQ.5.OR.NPRT(18).EQ.13) GO TO 161                   VARTTH
      IF (NPRT(18).EQ.10.OR.NPRT(18).EQ.11) GO TO 161                  VARTTH
      IF (NPRT(18).GE.15) GO TO 161                                    VARTTH
      DO  65  J=1,NSEG                                                 HEDING
      IF (MNSEG(J).EQ.0) GO TO 65                                      HEDING
      LSEG = MNSEG(J)                                                  HEDING
      DO 64  I=1,LSEG                                                  HEDING
      MSSF = MSSF+1                                                    HEDING
      NOPL(MSSF) = J                                                   HEDING
   64 MOPL(MSSF) = MSEG(2,I,J)                                         HEDING
   65 CONTINUE                                                         HEDING
      IF (MSSF.EQ.0) GO TO 70                                          HEDING
      DO 67 J=1,MSSF                                                   HEDING
      MT = MT + 1                                                      HEDING
      NT = MT                                                          HEDING
      IF  (LNEW)  NT = 6                                               HEDING
C     P & E CARRIAGE CONTROL                                           PECONV
      CALL CARCON(NT,1)                                                PECONV
      IT = MT - 20                                                     HEDING
      PAGE = FLOAT(MT) + XPAGE                                         HEDING
      IF (NT.EQ.6) WRITE(NT,121) DATE,BLANK,NPG                        PAGE
      IF (NT.NE.6) WRITE(NT,121) DATE                                  PAGE
      IF (NT.EQ.6) NPG=NPG+1                                           PAGE
      WRITE (NT,21) COMENT,PAGE,VPSTTL,BDYTTL                          PAGE
      N1 = NOPL(J)                                                     HEDING
      M1 = MOPL(J)                                                     HEDING
      WRITE (NT,68) N1,SEG(N1),M1,SEG(M1),UNITL,N1,M1                  HEDING
     *              ,UNITL,UNITM,UNITM,UNITM                           HEDING
      IF  (.NOT.LNEW)  GO TO 67                                        HEDING
      DO  66  I=1,NLINES                                               HEDING
   66 WRITE (NT, 69)  USEC(I),(ZTTH(JJ,I,IT),JJ=1,10)                  HEDING
   67 CONTINUE                                                         HEDING
   68 FORMAT('0',26X,'CONTACT FORCES - SEGMENT NO.',I3,' (',A4,        HEDING
     *                          ') VS. SEGMENT NO.',I3,' (',A4,')'//   HEDING
     *        13X,'DEFL-  NORMAL  FRICTION RESULTANT',                 HEDING
     *        14X,'CONTACT LOCATION (',A4,')'/                         HEDING
     *        4X,'TIME    ECTION',3(3X,'FORCE',1X),                    HEDING
     *        2('  SEG.',I3,' LOCAL REFERENCE ')/                      HEDING
```

```
     *       3X,'(MSEC)',3X,'(',A4,')', 3(3X,'(',A4,')'),                HEDING
     *       2(5X,'X',7X,'Y',7X,'Z',4X)/1X)                              HEDING
   69 FORMAT(2F9.3,3F9.2,3F8.3,2X,3F8.3)                                 HEDING
  161 CONTINUE                                                           VARTTH
C                                                                        HEDING
C     AIRBAG FORCES HEADINGS                                             HEDING
C                                                                        HEDING
   70 IF (NBAG.EQ.0) GO TO 82                                            HEDING
      IF (NPRT(18).EQ.6.OR.NPRT(18).EQ.9) GO TO 82                       VARTTH
      IF (NPRT(18).GE.12) GO TO 82                                       VARTTH
      DO  77  J=1,NBAG                                                   HEDING
      IF (MNBAG(J).EQ.0) GO TO 77                                        HEDING
      MT = MT + 1                                                        HEDING
      NT = MT                                                            HEDING
      IF  (LNEW)  NT = 6                                                 HEDING
C     P & E CARRIAGE CONTROL                                             PECONV
      CALL CARCON(NT,1)                                                  PECONV
      IT = MT - 20                                                       HEDING
      PAGE = FLOAT(MT) + XPAGE                                           HEDING
      IF (NT.EQ.6) WRITE(NT,121) DATE,BLANK,NPG                          PAGE
      IF (NT.NE.6) WRITE(NT,121) DATE                                    PAGE
      IF (NT.EQ.6) NPG=NPG+1                                             PAGE
      WRITE (NT,21) COMENT,PAGE,VPSTTL,BDYTTL                            PAGE
      WRITE (NT,78) J,(BAGTTL(I,J),I=1,5)                                HEDING
      IF  (.NOT.LNEW)  GO TO 72                                          HEDING
      DO  71  I=1,NLINES                                                 HEDING
   71 WRITE (NT, 79)  USEC(I),(ZTTH(JJ,I,IT),JJ=1,12)                    HEDING
   72 KBAG = 0                                                           HEDING
      KP = NPANEL(J)+1                                                   HEDING
      DO 73 K=1,KP                                                       HEDING
      KBAG = KBAG+1                                                      HEDING
   73 HEAD(KBAG) = PHED(K)                                               HEDING
      KP = MNBAG(J)                                                      HEDING
      DO 74 K=1,KP                                                       HEDING
      KBAG = KBAG+1                                                      HEDING
      M = MBAG(2,K,J)                                                    HEDING
   74 HEAD(KBAG) = SEG(M)                                                HEDING
      DO 76 J1=1,KBAG,4                                                  HEDING
      J2 = MIN0(J1+3,KBAG)                                               HEDING
      MT = MT + 1                                                        HEDING
      NT = MT                                                            HEDING
      IF  (LNEW)  NT = 6                                                 HEDING
C     P & E CARRIAGE CONTROL                                             PECONV
      CALL CARCON(NT,1)                                                  PECONV
      IT = MT - 20                                                       HEDING
      PAGE = FLOAT(MT) + XPAGE                                           HEDING
      IF (NT.EQ.6) WRITE(NT,121) DATE,BLANK,NPG                          PAGE
      IF (NT.NE.6) WRITE(NT,121) DATE                                    PAGE
      IF (NT.EQ.6) NPG=NPG+1                                             PAGE
      WRITE (NT,21) COMENT,PAGE,VPSTTL,BDYTTL                            PAGE
```

```
      WRITE (NT,80)UNITM,J,(BAGTTL(I,J),I=1,5),(BLANK,J,HEAD(K),K=J1,J2)HEDING
      WRITE (NT,51) (BLANK,K=J1,J2)                                     HEDING
      WRITE (NT,38)                                                     HEDING
      IF  (.NOT.LNEW)  GO TO 76                                         HEDING
      JJ = 3*(J2-J1+1)                                                  HEDING
      DO  75  I=1,NLINES                                                HEDING
   75 WRITE (NT, 81)  USEC(I),(ZTTH(K,I,IT),K=1,JJ)                     HEDING
   76 CONTINUE                                                          HEDING
   77 CONTINUE                                                          HEDING
   78 FORMAT('0',26X,'PARAMETERS FOR AIRBAG NO.',I2,4X,5A4//            HEDING
     *         16X,'SUPPLY CYLINDER   STATIC'/                          HEDING
     *         4X,'TIME',8X,'PRES.',4X,'TEMP.',4X,'PRES.',12X,'AIRBAG', HEDING
     *  3X,'CENTER',14X,'AIRBAG  SEMIAXES',12X,'ORIENTATION (DEG.)'/    HEDING
     *  3X,'(MSEC)',7X,'(PSIG)  (DEG.R)   (PSIG)',8X,'X',8X,'Y',8X,'Z', HEDING
     *  11X,'A',8X,'B',8X,'C',10X,'YAW',4X,'PITCH',5X,'ROLL'/ )         HEDING
   79 FORMAT (F9.3,3X,3F9.2,2(3X,3F9.3),3X,3F9.2)                       HEDING
   80 FORMAT('0',26X,'CONTACT FORCES (',A4,') ON AIRBAG NO.',I2,4X,5A4//HEDING
     *      /4X,'TIME',4(A1,11X,'AIRBAG',I2,' VS. ',A4,1X))             HEDING
   81 FORMAT (F9.3,4(3X,3F9.2))                                         HEDING
   82 RETURN                                                            HEDING
      END                                                               HEDING
```

```
      SUBROUTINE HERRON(HD3,NT1,THETO,THETOP)                           HERRON
C                                                 REV IV    07/23/86TWOPI
C     COMPUTES THETO - ANGLE OF JOINT STOP                              HERRON
C              THETOP- DERIVATIVE OF THETO WITH RESPECT TO PHI          HERRON
C                                                                       HERRON
C         FROM HD3   - COMPONENTS OF VECTOR DEFINING PHI                HERRON
C              NT1   - INDEX TO TAB ARRAY DEFINING FUNCTION             HERRON
C                                                                       HERRON
      IMPLICIT REAL*8(A-H,O-Z)                                          HERRON
      COMMON/TABLES/MXNTI,MXNTB,MXTB1,MXTB2,NTI(50),NTAB(1250),TAB(4500)DIMENB
      COMMON/CNSNTS/ PI,RADIAN,G,THIRD,EPS(24),                         HERRON
     *               UNITL,UNITM,UNITT,GRAVTY(3),TWOPI                  TWOPI
      DIMENSION HD3(3)                                                  HERRON
      IF (TAB(NT1+1).LE.0.0) GO TO 30                                   HERRON
      IF (TAB(NT1+2).LE.0.0) GO TO 30                                   HERRON
C                                                                       HERRON
C     THETO = P1(CP) + SP*P2(CP)                                        HERRON
C                                                                       HERRON
C     THETOP = -SP*P1'(CP) + CP*P2(CP) - SP**2*P2'(CP)                  HERRON
C                                                                       HERRON
C         WHERE P1(X),P2(X) ARE THE TWO 5TH ORDER POLYNOMIALS DEFINED   HERRON
C                           IN TAB(NT1+5) AND TAB(NT1+11)               HERRON
C               P1'(X),P2'(X) ARE THEIR DERIVATIVES WITH RESPECT TO PHI HERRON
C           AND SP,CP ARE SIN(PHI) AND COS(PHI)                         HERRON
C                                                                       HERRON
      STH2 = 1.0-HD3(3)**2                                              HERRON
      STH  = DSQRT(STH2)                                                HERRON
      CP   = HD3(1)/STH                                                 HERRON
      SP   = HD3(2)/STH                                                 HERRON
      P1   = TAB(NT1+5 )+ CP*(TAB(NT1+6 )                               HERRON
     *                  + CP*(TAB(NT1+7 )                               HERRON
     *                  + CP*(TAB(NT1+8 )                               HERRON
     *                  + CP*(TAB(NT1+9 )                               HERRON
     *                  + CP*(TAB(NT1+10) )))))                         HERRON
      P2   = TAB(NT1+11)+ CP*(TAB(NT1+12)                               HERRON
     *                  + CP*(TAB(NT1+13)                               HERRON
     *                  + CP*(TAB(NT1+14)                               HERRON
     *                  + CP*(TAB(NT1+15)                               HERRON
     *                  + CP*(TAB(NT1+16) )))))                         HERRON
      P1P  = TAB(NT1+6 )+ CP*(2.0*TAB(NT1+7 )                           HERRON
     *                  + CP*(3.0*TAB(NT1+8 )                           HERRON
     *                  + CP*(4.0*TAB(NT1+9 )                           HERRON
     *                  + CP*(5.0*TAB(NT1+10) ))))                      HERRON
      P2P  = TAB(NT1+12)+ CP*(2.0*TAB(NT1+13)                           HERRON
     *                  + CP*(3.0*TAB(NT1+14)                           HERRON
     *                  + CP*(4.0*TAB(NT1+15)                           HERRON
     *                  + CP*(5.0*TAB(NT1+16) ))))                      HERRON
      THETO  = P1 + SP*P2                                               HERRON
      THETOP = CP*P2 - SP*(P1P + SP*P2P)                                HERRON
      GO TO 99                                                          HERRON
```

```
C                                                                       HERRON
C     EVALUATE THETO AND THETOP FROM REGULAR FUNCTION DEFINITION WHERE  HERRON
C     THETO (ORDINATE) IS A FUNCTION OF PHI (ABSCISSA)  (0 < PHI < 2*PI)HERRON
C                                                                       HERRON
   30 PHI = DATAN2(HD3(2),HD3(1))                                       HERRON
      IF (PHI.LT.0.0) PHI = PHI + TWOPI                                 TWOPI
      THETO  = EVALFD(PHI,NT1,1)                                        HERRON
      THETOP = EVALFD(PHI,NT1,0)                                        HERRON
   99 RETURN                                                            HERRON
      END                                                               HERRON
```

```
      SUBROUTINE HICCSI(NPTS)                                            HICCSI
C                                                   REV IV    10/08/87PLTINC
C                                                                        HICCSI
C     COMPUTES HIC, HSI AND CSI FOR CVS PROGRAM.                         HICCSI
C                                                                        HICCSI
C     ASSUMES Z ARRAY CONTAINS                                           HICCSI
C     Z(I,1 ),I=1,NPTS : TIME POINTS (SECONDS)                           HICCSI
C     Z(I,JH),I=1,NPTS :  HEAD RESULTANT ACCELERATIONS (G'S)             HICCSI
C     Z(I,JC),I=1,NPTS : CHEST RESULTANT ACCELERATIONS (G'S)             HICCSI
C                                                                        HICCSI
C     NOTE:                                                              HICCSI
C     IF JDTPTS(1)=0,  HEAD RESULTANT IS NOT AVAILABLE (JH=NULL,JC=2).   HICCSI
C     IF JDTPTS(2)=0, CHEST RESULTANT IS NOT AVAILABLE (JH=2,JC=NULL).   HICCSI
C     OTHERWISE, JH=2 AND JC=3.                                          HICCSI
C                                                                        HICCSI
      COMMON/CDINT/ JDTPTS(18),Z(1000,3)                                 PLTINC
      COMMON/CONTRL/ TIME,NSEG,NJNT,NPL,NBLT,NBAG,NVEH,NGRND,            PAGE
     *               NS,NQ,NSD,NFLX,NHRNSS,NWINDF,NJNTF,NPRT(36),NPG     PAGE
      REAL*8 TIME                                                        PAGE
      DIMENSION AREA(1000)                                               PLTINC
      IF  (NPTS.LT.25)  GO TO 25                                         HICCSI
      WRITE  (6,14) NPG                                                  PAGE
      NPG=NPG+1                                                          PAGE
   14 FORMAT (1H1, '   HIC, HSI AND CSI RESULTS',96X,'PAGE',I5)          PAGE
      JH  = 2                                                            HICCSI
      JC  = 3                                                            HICCSI
      IF  (JDTPTS(1).EQ.0)  JC = 2                                       HICCSI
      CSI = 0.0                                                          HICCSI
      HSI = 0.0                                                          HICCSI
      HIC = 0.0                                                          HICCSI
      CMX = Z(1,JC)                                                      HICCSI
      HMX = Z(1,JH)                                                      HICCSI
      IF  (JDTPTS(2).EQ.0)  GO TO 16                                     HICCSI
C                                                                        HICCSI
C     COMPUTE CSI - CHEST SEVERITY INDEX                                 HICCSI
C                                                                        HICCSI
      H1  = SQRT(Z(1,JC)) * Z(1,JC)**2                                   HICCSI
      DO  15  I=2,NPTS                                                   HICCSI
      H2  = SQRT(Z(I,JC)) * Z(I,JC)**2                                   HICCSI
      DT  = Z(I,1) - Z(I-1,1)                                            HICCSI
      CSI = CSI + 0.5*DT*(H1+H2)                                         HICCSI
      IF  (CMX.GT.Z(I,JC))  GO TO 15                                     HICCSI
      CMX = Z(I,JC)                                                      HICCSI
      CMT = Z(I,1)                                                       HICCSI
   15 H1  = H2                                                           HICCSI
      CSI = 0.001*CSI                                                    HICCSI
   16 IF  (JDTPTS(1).EQ.0)  GO TO 23                                     HICCSI
C                                                                        HICCSI
C     COMPUTE HSI - HEAD SEVERITY INDEX - AND AREA TABLE                 HICCSI
C                                                                        HICCSI
```

```
      AREA(1) = 0.0                                                     HICCSI
      H1  = SQRT(Z(1,JH)) * Z(1,JH)**2                                  HICCSI
      DO  17  I=2,NPTS                                                  HICCSI
      H2  = SQRT(Z(I,JH)) * Z(I,JH)**2                                  HICCSI
      DT  = 0.5*(Z(I,1) - Z(I-1,1))                                     HICCSI
      AREA(I) = AREA(I-1) + DT*(Z(I-1,JH)+Z(I,JH))                      HICCSI
      HSI = HSI + DT*(H1+H2)                                            HICCSI
      IF  (HMX.GT.Z(I,JH))  GO TO 17                                    HICCSI
      HMX = Z(I,JH)                                                     HICCSI
      HMT = Z(I,1)                                                      HICCSI
   17 H1  = H2                                                          HICCSI
      HSI = 0.001*HSI                                                   HICCSI
C                                                                       HICCSI
C      COMPUTE HIC - HEAD INJURY CRITERION - AND TIME DURATION HT1,HT2  HICCSI
C                                                                       HICCSI
      DO  19  K=2,NPTS                                                  HICCSI
      DO  18  L=K,NPTS                                                  HICCSI
      DT  = Z(L,1)  -  Z(K-1,1)                                         HICCSI
      DH  = AREA(L) - AREA(K-1)                                         HICCSI
      HT  = DH/DT                                                       HICCSI
      HM  = DT*SQRT(HT)*HT**2                                           HICCSI
      IF  (HM.LE.HIC)  GO TO 18                                         HICCSI
      HIC = HM                                                          HICCSI
      HT1 = Z(K-1,1)                                                    HICCSI
      HT2 = Z(L,1)                                                      HICCSI
      HA2 = Z(L,JH)                                                     HICCSI
      HA1 = Z(K-1,JH)                                                   HICCSI
      AVE = HT                                                          HICCSI
   18 CONTINUE                                                          HICCSI
   19 CONTINUE                                                          HICCSI
      HIC = 0.001*HIC                                                   HICCSI
      WRITE  (6,21)  HIC,HT1,HT2,HA1,HA2,AVE                            HICCSI
   21 FORMAT (1H0, '   HEAD INJURY CRITERION'//                         HICCSI
     *          '    HIC = ', F8.2,                                     HICCSI
     *              9X, 'TIME DURATION = ', F9.3, '  TO ', F9.3, ' MSEC'/ HICCSI
     *    20X, 'WITH HEAD RESULTANTS = ', F9.3, ' AND ', F9.3, ' G''S'//HICCSI
     *14X,'AVERAGE HEAD RESULTANT FOR TIME DURATION = ', F9.3, ' G''S') HICCSI
      WRITE  (6,22)  HSI,HMX,HMT                                        HICCSI
   22 FORMAT (1H0, '   HEAD SEVERITY INDEX'//                           HICCSI
     *          '    HSI = ', F8.2//                                    HICCSI
     *    '     MAX  HEAD RESULTANT = ', F9.3, ' G''S AT ', F9.3, ' MSEC')HICCSI
   23 IF  (JDTPTS(2).EQ.0)  GO TO 25                                    HICCSI
      WRITE  (6,24)  CSI,CMX,CMT                                        HICCSI
   24 FORMAT (1H0, '   CHEST SEVERITY INDEX'//                          HICCSI
     *          '    CSI = ', F8.2//                                    HICCSI
     *    '     MAX CHEST RESULTANT = ', F9.3, ' G''S AT ', F9.3, ' MSEC')HICCSI
   25 CONTINUE                                                          TGMOD1
      IF(NPTS.LT.25) WRITE(6,101) NPTS                                  TGMOD1
  101 FORMAT(1X,//,2X,'HIC, HSI AND CSI NOT COMPUTED BECAUSE THE NUMBER TGMOD1
     *OF POINTS TO BE USED IN THE COMPUTATION =',I2,',',/,2X            TGMOD1
```

```
     * 'WHICH IS LESS THAN THE MINIMUM REQUIREMENT OF 25 POINTS.',/)     TGMOD1
      RETURN                                                             TGMOD1
      END                                                                HICCSI
```

```
      SUBROUTINE HINPUT                                                 HINPUT
C                                               REV IV    07/23/86TWOPI
C     CONTROLS THE INPUT OF CARDS F.8.A - F.8.D CONTAINING THE SETUP ANDHINPUT
C     CONTROL OF THE HARNESS BELT SYSTEM.                               HINPUT
C                                                                       HINPUT
C                                                                       HINPUT
      IMPLICIT REAL*8(A-H,O-Z)                                          HINPUT
      COMMON/CONTRL/ TIME,NSEG,NJNT,NPL,NBLT,NBAG,NVEH,NGRND,           HINPUT
     *               NS,NQ,NSD,NFLX,NHRNSS,NWINDF,NJNTF,NPRT(36),NPG    PAGE
      COMMON/CNSNTS/ PI,RADIAN,G,THIRD,EPS(24),                         HINPUT
     *               UNITL,UNITM,UNITT,GRAVTY(3),TWOPI                  TWOPI
      COMMON/HRNESS/ BAR(15,100),BB(100),BBDOT(100),PLOSS(2,100),       HINPUT
     *               XLONG(20),HTIME(2),IBAR(5,100),NL(2,100),          HINPUT
     *               NPTSPB(20),NPTPLY(20),NTHRNS(20),NBLTPH(5)         HINPUT
      COMMON/TABLES/MXNTI,MXNTB,MXTB1,MXTB2,NTI(50),NTAB(1250),TAB(4500)DIMENB
      COMMON/CNTSRF/ PL(24,30),BELT(20,8),TPTS(6,8),BD(24,40)           EDGE
      COMMON/TITLES/ DATE(3),COMENT(40),VPSTTL(20),BDYTTL(5),           HINPUT
     *               BLTTTL(5,8),PLTTL(5,30),BAGTTL(5,6),SEG(30),       HINPUT
     *               JOINT(30),CGS(30),JS(30)                           HINPUT
      REAL DATE,COMENT,VPSTTL,BDYTTL,BLTTTL,PLTTL,BAGTTL,SEG,JOINT      HINPUT
      LOGICAL*1 CGS,JS                                                  HINPUT
C     THIS COMMON/TEMTVS/ IS SHARED BY CINPUT, FINPUT, HINPUT AND FDINITHINPUT
      COMMON/TEMPVS/ JTITLE(5,51),NF(5),MS(3),KTITLE(31)                HINPUT
      REAL JTITLE,KTITLE                                                HINPUT
      IF  (NHRNSS.EQ.0)  GO TO 99                                       HINPUT
C                                                                       HINPUT
C     INPUT CARD F.8.A                                                  HINPUT
C         (NOTE: NHRNSS NOW SUPPLIED ON INPUT CARD D.1)                 HINPUT
C         NBLTPH - NO. OF BELTS PER HARNESS                             HINPUT
C                                                                       HINPUT
      READ  (5,11)  (NBLTPH(I),I=1,NHRNSS)                              HINPUT
   11 FORMAT(18I4)                                                      HINPUT
      WRITE (6,12) NPG,NHRNSS,(NBLTPH(I),I=1,NHRNSS)                    PAGE
      NPG=NPG+1                                                         PAGE
   12 FORMAT('1 HARNESS-BELT SYSTEM INPUT',96X,'PAGE',I5/120X,          PAGE
     *       'CARDS F.8'/'  NO. OF HARNESSES =',I4//                    PAGE
     *        '  NO. OF BELTS PER HARNESS =',5I6)                       HINPUT
      J1 = 1                                                            HINPUT
      K1 = 1                                                            HINPUT
      DO 90 I=1,NHRNSS                                                  HINPUT
      IF (NBLTPH(I).LE.0) GO TO 90                                      HINPUT
      J2 = J1 + NBLTPH(I) -1                                            HINPUT
C                                                                       HINPUT
C     INPUT CARD F.8.B - NPTSPB - NO. OF POINTS PER BELT.               HINPUT
C                                                                       HINPUT
      READ (5,11) (NPTSPB(J),J=J1,J2)                                   HINPUT
      WRITE (6,13) I,(NPTSPB(J),J=J1,J2)                                HINPUT
   13 FORMAT('0 FOR HARNESS NO.',I3,' NO. OF POINTS PER BELT =',20I4)   HINPUT
      DO 80 J=J1,J2                                                     HINPUT
      IF (NPTSPB(J).EQ.0) GO TO 80                                      HINPUT
```

```
C                                                                       HINPUT
C     INPUT CARD F.8.C - 5 FUNCTION NOS AND LENGTH OF EACH BELT.        HINPUT
C                                                                       HINPUT
      READ  (5,14)     NF,XLONG(J)                                      HINPUT
   14 FORMAT(5I4,F12.6)                                                 HINPUT
      WRITE (6,15) I,J,NF,XLONG(J),UNITL                                HINPUT
   15 FORMAT('0 HARNESS NO.',I3,' BELT NO.',I3,' FUNCTION NOS.',5I6,    HINPUT
     *       '  REFERENCE SLACK = ',F9.3,1X,A4/)                        HINPUT
      IF  (XLONG(J).EQ.0.0)  XLONG(J) = EPS(24)                         HINPUT
      WRITE  (6,16)                                                     HINPUT
   16 FORMAT ('0    K    KS    KE    NT    NPD   NDR    FUNCTION NOS.', HINPUT
     *        66X,'CARDS F.8.D'/)                                       CHGIII
C                                                                       HINPUT
C     SET UP POINTERS IN NTAB AND INITIAL VALUES OF TAB FOR BELT J      HINPUT
C         AS WAS DONE FOR OTHER CONTACTS IN SUBROUTINE FINPUT.          HINPUT
C                                                                       HINPUT
      NTHRNS(J) = MXNTB+1                                               HINPUT
      CALL FDINIT                                                       HINPUT
      K2 = K1 + NPTSPB(J) - 1                                           HINPUT
      DO 70 K=K1,K2                                                     HINPUT
C                                                                       HINPUT
C     INPUT CARD F.8.D                                                  HINPUT
C                                                                       HINPUT
      READ  (5,21)  KS,KE,NPD,NDR,NF, (BAR(L,K),L=1,3)                  HINPUT
   21 FORMAT (9I4,3F12.0)                                               HINPUT
      READ  (5,22)  (BAR(L,K),L=7,12)                                   HINPUT
   22 FORMAT (6F12.0)                                                   HINPUT
      ICHEC = 0                                                         CHGIII
      IF (K.EQ.K1.OR.K.EQ.K2) ICHEC = 1                                 CHGIII
      IF (ICHEC.EQ.1.AND.NPD.EQ.0) STOP 60                              CHGIII
      IF (ICHEC.EQ.1.AND.NDR.EQ.0) STOP 61                              CHGIII
      IF (NDR.EQ.0.AND.NPD.NE.0) STOP 62                                CHGIII
      IBAR(1,K) = KS                                                    HINPUT
      IBAR(2,K) = KE                                                    HINPUT
      IBAR(4,K) = NPD                                                   HINPUT
      IBAR(5,K) = NDR                                                   HINPUT
      IBAR(3,K) = MXNTB+1                                               HINPUT
      CALL FDINIT                                                       HINPUT
      SQRER = 1.0                                                       HINPUT
      IF  (KE.NE.0)  SQRER = DSQRT(XDY(BAR(1,K),BD(7,KE),BAR(1,K)))     HINPUT
      DO  26  L=1,3                                                     HINPUT
      IF  (KE.NE.0)  BAR(L+6,K) = BD(L+3,KE)                            HINPUT
   26 BAR(L+3,K) = BAR(L,K)/SQRER                                       HINPUT
      WRITE  (6,31)  K,(IBAR(L,K),L=1  ),NF                             HINPUT
   31 FORMAT (11I6)                                                     HINPUT
   70 CONTINUE                                                          HINPUT
      WRITE  (6,71)  UNITL,UNITL,UNITL,UNITL                            HINPUT
   71 FORMAT ('0',12X,'BASE REFERENCE (',      A4,')',                  HINPUT
     *             7X,'ADJUSTED REFERENCE (', A4,')',                   HINPUT
     *            11X,'OFFSET (',              A4,')',                  HINPUT
```

```
     *           11X,'PREFERRED DIRECTION (',A4,')'/                          HINPUT
     *       5X,'K', 4(8X,'X',8X,'Y',8X,'Z',3X) /)                            HINPUT
      WRITE  (6,72)  (K,(BAR(L,K),L=1,12),K=K1,K2)                            HINPUT
   72 FORMAT (I6,3X,3F9.3,3X,3F9.3,3X,3F9.3,3X,3F9.3)                         HINPUT
      K1 = K2+1                                                               HINPUT
   80 CONTINUE                                                                HINPUT
      J1 = J2+1                                                               HINPUT
   90 CONTINUE                                                                HINPUT
      DO  92  K=1,100                                                         HINPUT
      BBDOT(K) = 0.0                                                          HINPUT
      DO  91  J=1,2                                                           HINPUT
   91 PLOSS(J,K) = 0.0                                                        HINPUT
      DO  92  J=1,3                                                           HINPUT
   92 BAR(J+12,K) = 0.0                                                       HINPUT
   99 RETURN                                                                  HINPUT
      END                                                                     HINPUT
```

```
      SUBROUTINE HPTURB                                                 HPTURB
C                                                  REV IV    07/23/86TWOPI
      IMPLICIT REAL*8 (A-H,O-Z)                                         HPTURB
      COMMON/CONTRL/ TIME,NSEG,NJNT,NPL,NBLT,NBAG,NVEH,NGRND,           HPTURB
     *               NS,NQ,NSD,NFLX,NHRNSS,NWINDF,NJNTF,NPRT(36),NPG    PAGE
      COMMON/CNSNTS/ PI,RADIAN,G,THIRD,EPS(24),                         HPTURB
     *               UNITL,UNITM,UNITT,GRAVTY(3),TWOPI                  TWOPI
      COMMON/CNTSRF/ PL(24,30),BELT(20,8),TPTS(6,8),BD(24,40)           EDGE
      COMMON/SGMNTS/ D(3,3,30),WMEG(3,30),WMEGD(3,30),U1(3,30),U2(3,30),HPTURB
     *               SEGLP(3,30),SEGLV(3,30),SEGLA(3,30),NSYM(30)       HPTURB
      COMMON/RSAVE/  XSG(3,20,3),DPMI(3,3,30),LPMI(30),                 ATBIII
     *               NSG(9),MSG(20,9),MCG,MCGIN(24,5),KREF(20,9)        TTHKREF
      COMMON/HRNESS/ BAR(15,100),BB(100),BBDOT(100),PLOSS(2,100),       HPTURB
     *               XLONG(20),HTIME(2),IBAR(5,100),NL(2,100),          HPTURB
     *               NPTSPB(20),NPTPLY(20),NTHRNS(20),NBLTPH(5)         HPTURB
C     THIS COMMON/TEMPVS/ IS SHARED BY HPTURB, HBPLAY, HBELT AND HSETC. HPTURB
      COMMON/TEMPVS/ B(3,3,3),S(3,3),T(3),R(3),V(3),T1(3),T2(3),        HPTURB
     *               E(3,3,50),EDOT(3,50),FCE(3,50),FR(3,50),ZR(3,50),  HPTURB
     *               TR(3,50),U(3,50),PTLOSS(2,50),BL(50),FB(50),FP(50),HPTURB
     *               OLDBB(100),RHS(3,54),C(3,3,200),IJK(54,54)         HPTURB
      DIMENSION BLOSS(2,20),HLOSS(2,5)                                  HPTURB
      EQUIVALENCE (BLOSS(1,1),C(1,1,1)) , (HLOSS(1,1),C(1,1,10))        HPTURB
      LOGICAL LAST                                                      HPTURB
      DATA MAXITR/10/                                                   HPTURB
      CALL ELTIME (1,39)                                                HPTURB
      CALL HBPLAY                                                       HPTURB
      DHT = 0.0                                                         HPTURB
      IF  (TIME.NE.0.0)  DHT = TIME - HTIME(1)                          HPTURB
      HTIME(1) = TIME                                                   HPTURB
      DO  11  J=1,100                                                   HPTURB
      PTLOSS(J,1) = 0.0                                                 HPTURB
      OLDBB(J) = BB(J)                                                  HPTURB
      DO  11  I=1,3                                                     HPTURB
   11 BAR(I,J) = BAR(I+3,J)                                             HPTURB
      TSEC = 1000.0*TIME                                                HPTURB
      IF  (NPRT(28).NE.0)  WRITE (6,12) TSEC,NPG,UNITL,UNITM,UNITL,     PAGE
     *                                  UNITL,UNITM,UNITL,UNITM         HPTURB
      IF  (NPRT(28).NE.0) NPG=NPG+1                                     PAGE
   12 FORMAT('1   HARNESS BELT RESULTS FOR TIME =',F9.3,' MSEC.',73X,   PAGE
     *       'PAGE',I5///                                               PAGE
     *       36X,'BELT STRAIN',6X,'(LOCAL OR ELLIPSOID)',18X,           CHGIII
     *       '(INERTIAL)',14X,'PENETRATION'/                            CHGIII
     *       '      POINT   POINT  SEGMENT LENGTH  ENERGY LOSS',5X,     HPTURB
     *       'REFERENCE POINT (',A4,')',13X,'BELT FORCES  (',A4,')',    AFREVS
     *       9X,'ENERGY LOSS'/                                          HPTURB
     *       '       NO.    INDEX    NO.    (',A4,')  (',2A4,')',7X,    HPTURB
     *       'X',8X,'Y',8X,'Z',13X,'X',10X,'Y',10X,'Z',8X,'(',2A4,')'/) HPTURB
      J1 = 1                                                            HPTURB
      K0 = 1                                                            HPTURB
      KNL0 = 0                                                          HPTURB
```

```
      DO  61  NH=1,NHRNSS                                              HPTURB
      IF  (NBLTPH(NH).LE.0)  GO TO 61                                  HPTURB
      ITER = 1                                                         HPTURB
      KNL1 = KNLO                                                      HPTURB
      KNLN = 0                                                         CHGIII
C                                                                      HPTURB
C     START OF   DO  59  ITER=1,MAXITR   LOOP                          HPTURB
C                                                                      HPTURB
   13 NJ2 = 54                                                         HPTURB
      DO  14  I=1,NJ2                                                  HPTURB
      DO  14  J=1,NJ2                                                  HPTURB
   14 IJK(I,J) = 0                                                     HPTURB
      KNLO = KNL1                                                      HPTURB
      J2   = J1 + NBLTPH(NH) - 1                                       HPTURB
      NTP  = 0                                                         HPTURB
      IJ   = 0                                                         HPTURB
      CALL HBELT (J1,J2,KNLO,1)                                        HPTURB
      KHO  = 0                                                         HPTURB
      KNLO = KNL1                                                      HPTURB
      DO  15  NB=J1,J2                                                 HPTURB
      IF  (NPTPLY(NB).LE.0)  GO TO 15                                  HPTURB
      NPTS = NPTPLY(NB)                                                HPTURB
      CALL HSETC (NPTS,KHO,KNLO,NTP,IJ)                                HPTURB
      KHO = KHO + NPTS                                                 HPTURB
      KNLO = KNLO + NPTS                                               HPTURB
   15 CONTINUE                                                         HPTURB
      KNLN = KNLO                                                      CHGIII
C                                                                      HPTURB
C     SET UP C AND IJK ELEMENTS FOR TIE-POINTS.                        HPTURB
C                                                                      HPTURB
      KNLO = KNL1                                                      HPTURB
      KNLK = KNLO + 1                                                  HPTURB
      K1   = KNLK                                                      HPTURB
      DO  22  NB=J1,J2                                                 HPTURB
      IF  (NPTPLY(NB).LE.0)  GO TO 22                                  HPTURB
      K2   = K1 + NPTPLY(NB) - 1                                       HPTURB
      DO  21  KNL=K1,K2                                                HPTURB
      KI   = NL(1,KNL)                                                 HPTURB
      KS   = IABS(IBAR(1,KI))                                          HPTURB
      IF  (KS.LT.100)  GO TO 21                                        HPTURB
      KS1  = KS/100                                                    HPTURB
      DO  16  K=KNLK,KNL                                               HPTURB
      KK   = K                                                         HPTURB
      KI   = NL(1,K)                                                   HPTURB
      KS   = IABS(IBAR(1,KI))                                          HPTURB
      IF  (KS.LT.100)  GO TO 16                                        HPTURB
      KS2  = KS/100                                                    HPTURB
      IF  (KS2.EQ.KS1)  GO TO 17                                       HPTURB
   16 CONTINUE                                                         HPTURB
   17 IF  (KK.EQ.KNL)  GO TO 21                                        HPTURB
```

```
      KK1 = KK  - KNLO                                                  HPTURB
      KK2 = KNL - KNLO                                                  HPTURB
      IQ1 = MAX0(1,KK2-1)                                               HPTURB
      IQ2 = MIN0(KK2+1,KHO)                                             HPTURB
      DO  18  IQ=IQ1,IQ2                                                HPTURB
      IF  (IJK(KK2,IQ).EQ.0)  GO TO 18                                  HPTURB
      IJK(KK1,IQ) = IJK(KK2,IQ)                                         HPTURB
      IJK(KK2,IQ) = 0                                                   HPTURB
   18 CONTINUE                                                          HPTURB
      IJK(KK2,KK2) = IJ+1                                               HPTURB
      IJK(KK2,KK1) = IJ+2                                               HPTURB
      DO  20  J=1,3                                                     HPTURB
      DO  19  I=1,3                                                     HPTURB
      C(I,J,IJ+1) =  0.0                                                HPTURB
   19 C(I,J,IJ+2) =  0.0                                                HPTURB
      C(J,J,IJ+1) =  1.0                                                HPTURB
   20 C(J,J,IJ+2) = -1.0                                                HPTURB
      IJ    = IJ + 2                                                    HPTURB
   21 CONTINUE                                                          HPTURB
      K1    = K2 + 1                                                    HPTURB
   22 CONTINUE                                                          HPTURB
      MJ2 = -(KHO+NTP)                                                  HPTURB
      IF  (NPRT(28).LT.3)  GO TO 29                                     HPTURB
      NJ2 = -MJ2                                                        HPTURB
      DO  25  J=1,NJ2                                                   HPTURB
   25 WRITE  (6,26)  J,(RHS(I,J),I=1,3),(IJK(J,I),I=1,NJ2)              HPTURB
   26 FORMAT (I6,3F12.6,20I4/(42X,20I4))                                HPTURB
      DO  27  KLM=1,IJ                                                  HPTURB
   27 WRITE  (6,28)  KLM,((C(J,I,KLM),I=1,3),J=1,3)                     HPTURB
   28 FORMAT (I6,9F12.6)                                                HPTURB
   29 CALL FSMSOL (C,RHS,IJK,MJ2,IJ,54,200)                             HPTURB
      IF  (NPRT(28).LT.3)  GO TO 31                                     HPTURB
      DO  30  J=1,NJ2                                                   HPTURB
   30 WRITE  (6,26)  J,(RHS(I,J),I=1,3),(IJK(J,I),I=1,NJ2)              HPTURB
   31 ONE = 1.0                                                         HPTURB
      DELMAX = 0.0                                                      HPTURB
      SCALE = 1.0                                                       HPTURB
      DO  44  IT=1,2                                                    HPTURB
      K1 = K0                                                           HPTURB
      KH = 0                                                            HPTURB
      KR = NTP                                                          HPTURB
      DO  43  NB=J1,J2                                                  HPTURB
      IF  (NPTPLY(NB).LE.0)  GO TO 43                                   HPTURB
      K2 = K1 + NPTPLY(NB) - 1                                          HPTURB
      DO  42  K=K1,K2                                                   HPTURB
      KH = KH + 1                                                       HPTURB
      KR = KR + 1                                                       HPTURB
C                                                                       HPTURB
C     HERE K IS INDEX OF ALL POINTS IN PLAY                             HPTURB
C         KH IS INDEX OF ALL POINTS IN PLAY ON A SINGLE HARNESS         HPTURB
```

```
C         KR IS INDEX OF RHS ARRAY ELEMENTS                                HPTURB
C                                                                          HPTURB
      KI = NL(1,K)                                                         HPTURB
      KS   = IABS(IBAR(1,KI))                                              HPTURB
      IF  (KS.GT.100)  KS = MOD(KS,100)                                    HPTURB
      IF  (IBAR(5,KI).EQ.0)  GO TO 32                                      HPTURB
      CALL MAT31 (D(1,1,KS),RHS(1,KR),R)                                   HPTURB
      GO TO 37                                                             HPTURB
C                                                                          HPTURB
C     NOTE: ENDPOINTS (K = K1 & K2) MUST BE TYPE 5.                        HPTURB
C                                                                          HPTURB
   32 CALL DOT31 (E(1,1,KH),RHS(1,KR),T1)                                  HPTURB
      IF  (IT.EQ.2)  GO TO 33                                              HPTURB
      DELMAX = DMAX1(DELMAX,DABS(T1(2)/DMIN1(BB(K),BB(K-1))))              HPTURB
      GO TO 34                                                             HPTURB
   33 BB(K  ) = BB(K  ) + SCALE*T1(2)                                      HPTURB
      BB(K-1) = BB(K-1) - SCALE*T1(2)                                      HPTURB
   34 DO  35  J=1,3                                                        HPTURB
   35 T2(J) = T1(1)*E(J,1,KH) + T1(3)*E(J,3,KH)                            HPTURB
      CALL MAT31 (D(1,1,KS),T2,R)                                          HPTURB
      IF  (NPRT(28).GE.3)  WRITE (6,36) K,T1,T2,R                          HPTURB
   36 FORMAT ('0',I6,3(3X,3F12.6))                                         HPTURB
   37 IF  (IT.EQ.2)  GO TO 39                                              HPTURB
      DO  38  J=1,3                                                        HPTURB
   38 DELMAX = DMAX1(DELMAX,DABS(R(J)/DMAX1(EPS(1),DABS(BAR(J+3,KI))))) HPTURB
      GO TO 42                                                             HPTURB
   39 DO  40  J=1,3                                                        HPTURB
   40 BAR(J+3,KI) = BAR(J+3,KI) + SCALE*R(J)                               HPTURB
      KE = IBAR(2,KI)                                                      HPTURB
      IF  (KE.EQ.0)  GO TO 42                                              HPTURB
      RER = XDY(BAR(4,KI),BD(7,KE),BAR(4,KI))                              HPTURB
      IF  (RER.LE.1.0)  GO TO 42                                           HPTURB
      SQRER = 1.0/DSQRT(RER)                                               HPTURB
      DO  41  J=1,3                                                        HPTURB
   41 BAR(J+3,KI) = SQRER*BAR(J+3,KI)                                      HPTURB
   42 CONTINUE                                                             HPTURB
      K1 = K2 + 1                                                          HPTURB
   43 CONTINUE                                                             HPTURB
      IF  (IT.EQ.2)  GO TO 44                                              HPTURB
      IF  (DELMAX.NE.0.0)  SCALE = DMIN1(ONE,EPS(1)/DELMAX)                HPTURB
   44 CONTINUE                                                             HPTURB
      IF  (NPRT(28).GE.2)  WRITE  (6,45)  ITER,DELMAX,SCALE                HPTURB
   45 FORMAT ('0  ITER =',I6,'  DELMAX =',F15.6,'  SCALE =',F15.6)         HPTURB
      LAST = DELMAX.LE.EPS(2) .OR. ITER.EQ.MAXITR                          HPTURB
      IF  (.NOT.LAST)  GO TO 52                                            HPTURB
      KH = 0                                                               HPTURB
      K1 = K0                                                              HPTURB
      HLOSS(1,NH) = 0.0                                                    HPTURB
      HLOSS(2,NH) = 0.0                                                    HPTURB
      DO  51  NB=J1,J2                                                     HPTURB
```

```
      BLOSS(1,NB) = 0.0                                                 HPTURB
      BLOSS(2,NB) = 0.0                                                 HPTURB
      IF  (NPTPLY(NB).LE.0)  GO TO 51                                   HPTURB
      K2 = K1 + NPTPLY(NB) - 1                                          HPTURB
      KK1 = NL(1,K1)                                                    HPTURB
      KK2 = NL(1,K2)                                                    HPTURB
      DO  46  K=KK1,KK2                                                 HPTURB
      DO  46  J=1,3                                                     HPTURB
   46 BAR(J+12,K) = 0.0                                                 HPTURB
      IF  (DHT.EQ.0.0)  GO TO 49                                        HPTURB
      DO  48  K=K1,K2                                                   HPTURB
      KH = KH + 1                                                       HPTURB
      KI = NL(1,K)                                                      HPTURB
      PLOSS(2,KI ) = PLOSS(2,KI ) + DHT*PTLOSS(2,KH)                    HPTURB
      IF  (K.EQ.K1)  GO TO 47                                           HPTURB
      BBDOT(K-1) = (BB(K-1)-OLDBB(K-1))/DHT                             HPTURB
      PLOSS(1,K-1) = PLOSS(1,K-1) + DHT*PTLOSS(1,KH-1)                  HPTURB
      BLOSS(1,NB) = BLOSS(1,NB) + PLOSS(1,K-1)                          HPTURB
   47 DO  48  J=1,3                                                     HPTURB
   48 BAR(J+12,KI) = (BAR(J+3,KI)-BAR(J,KI))/DHT                        HPTURB
      BBDOT(K2) = 0.0                                                   HPTURB
      PLOSS(1,K2) = 0.0                                                 HPTURB
   49 K1 = K2+1                                                         HPTURB
      DO  50  K=KK1,KK2                                                 HPTURB
   50 BLOSS(2,NB) = BLOSS(2,NB) + PLOSS(2,K)                            HPTURB
      HLOSS(1,NH) = HLOSS(1,NH) + BLOSS(1,NB)                           HPTURB
      HLOSS(2,NH) = HLOSS(2,NH) + BLOSS(2,NB)                           HPTURB
   51 CONTINUE                                                          HPTURB
   52 IF  (NPRT(28).EQ.0)  GO TO 59                                     HPTURB
      IF  (.NOT.LAST .AND. IABS(NPRT(28)).EQ.1)  GO TO 59               HPTURB
      K1 = K0                                                           HPTURB
      KH = 0                                                            HPTURB
      DO  57  NB=J1,J2                                                  HPTURB
      IF  (NPTPLY(NB).LE.0)  GO TO 57                                   HPTURB
      WRITE (6,53)  NB,NH                                               HPTURB
   53 FORMAT ('0  BELT NO.',I4,' OF HARNESS NO.',I4)                    HPTURB
      K2 = K1 + NPTPLY(NB) - 1                                          HPTURB
      DO  54  K=K1,K2                                                   HPTURB
      KH = KH + 1                                                       HPTURB
      KI = NL(1,K)                                                      HPTURB
      KS = IBAR(1,KI)                                                   HPTURB
      BK = 0.0                                                          HPTURB
      IF  (K.NE.K1)  BK = BB(K-1)                                       HPTURB
      PLS = 0.0                                                         HPTURB
      IF  (K.NE.K1)  PLS = PLOSS(1,K-1)                                 HPTURB
      T(1) = BAR(4,KI)                                                  HPTURB
      T(2) = BAR(5,KI)                                                  HPTURB
      T(3) = BAR(6,KI)                                                  HPTURB
      KJ = MOD(IABS(KS),100)                                            HPTURB
      IF  (LPMI(KJ).NE.0)  CALL DOT31 (DPMI(1,1,KJ),BAR(4,KI),T)        HPTURB
```

```
   54 WRITE (6,55) K,KI,KS,BK,PLS,(T(J),J=1,3),                          HPTURB
     *              (FCE(J,KH),J=1,3),PLOSS(2,KI)                         HPTURB
   55 FORMAT (3I8,F10.3,F12.3,2X,3F9.3,3X,3F11.3,3X,F12.3)               HPTURB
      IF  (LAST)  WRITE (6,56)  BLOSS(1,NB),BLOSS(2,NB)                   HPTURB
   56 FORMAT ('0    TOTAL BELT ENERGY LOSS',7X,F12.3,68X,F12.3)           HPTURB
      K1 = K2 + 1                                                         HPTURB
   57 CONTINUE                                                            HPTURB
      IF  (LAST)  WRITE (6,58)  HLOSS(1,NH),HLOSS(2,NH)                   HPTURB
   58 FORMAT ('0 TOTAL HARNESS ENERGY LOSS',7X,F12.3,68X,F12.3)           HPTURB
   59 ITER = ITER + 1                                                     HPTURB
C                                                                         HPTURB
C        END OF   DO  59  ITER=1,MAXITR   LOOP                            HPTURB
C                                                                         HPTURB
      IF  (.NOT.LAST)  GO TO 13                                           HPTURB
      IF  (ITER.GT.MAXITR)  WRITE  (6,60)  MAXITR,TSEC,DELMAX,SCALE       HPTURB
   60 FORMAT ('0 HPTURB ITER =',I4,' AT TIME =',F8.3,                     HPTURB
     *        ' MSEC.  DELMAX =',F10.6,'  SCALE =',F10.6)                 HPTURB
      J1 = J2 + 1                                                         HPTURB
      K0 = K1                                                             HPTURB
      KNLO = KNLN                                                         CHGIII
   61 CONTINUE                                                            HPTURB
      IF  (NPRT(28).LT.0)  NPRT(28) = 0                                   HPTURB
      CALL ELTIME (2,39)                                                  HPTURB
      RETURN                                                              HPTURB
      END                                                                 HPTURB
```

```
      SUBROUTINE HSETC (NPTS,KHO,KNLO,NTP,IJ)                          HSETC
C                                                 REV III.2 08/08/84REVIII
      IMPLICIT REAL*8 (A-H,O-Z)                                        HSETC
      COMMON/SGMNTS/ D(3,3,30),WMEG(3,30),WMEGD(3,30),U1(3,30),U2(3,30),HSETC
     *               SEGLP(3,30),SEGLV(3,30),SEGLA(3,30),NSYM(30)      HSETC
      COMMON/TABLES/MXNTI,MXNTB,MXTB1,MXTB2,NTI(50),NTAB(1250),TAB(4500)DIMENB
      COMMON/HRNESS/ BAR(15,100),BB(100),BBDOT(100),PLOSS(2,100),      HSETC
     *               XLONG(20),HTIME(2),IBAR(5,100),NL(2,100),         HSETC
     *               NPTSPB(20),NPTPLY(20),NTHRNS(20),NBLTPH(5)        HSETC
C     THIS COMMON/TEMPVS/ IS SHARED BY HPTURB, HBPLAY, HBELT AND HSETC. HSETC
      COMMON/TEMPVS/ B(3,3,3),S(3,3),T(3),R(3),V(3),T1(3),T2(3),       HSETC
     *               E(3,3,50),EDOT(3,50),FCE(3,50),FR(3,50),ZR(3,50), HSETC
     *               TR(3,50),U(3,50),PTLOSS(2,50),BL(50),FB(50),FP(50),HSETC
     *               OLDBB(100),RHS(3,54),C(3,3,200),IJK(54,54)        HSETC
      DIMENSION  KM(3),MK(2)                                           HSETC
      ONE = 1.0                                                        HSETC
      KNL = KNLO                                                       HSETC
      KH = KHO                                                         HSETC
      K1 = KHO + NTP + 1                                               HSETC
      K2 = KHO + NTP + NPTS                                            HSETC
      DO  60  K=K1,K2                                                  HSETC
C                                                                      HSETC
C     HERE K   IS INDEX OF IJK AND RHS ARRAYS                          HSETC
C          KH  IS INDEX OF POINTS IN PLAY ON EACH HARNESS              HSETC
C          KNL IS INDEX OF ALL POINTS IN PLAY                          HSETC
C          KI  IS INDEX OF ALL POINTS                                  HSETC
C                                                                      HSETC
      KH = KH + 1                                                      HSETC
      KNL = KNL + 1                                                    HSETC
C                                                                      HSETC
C     ZERO C(K,K) , C(K,K-1) , C(K,K+1) & RHS(K); SET IJK(K,K) = IJ    HSETC
C                                                                      HSETC
      KM(1) = K+1                                                      HSETC
      KM(2) = K-1                                                      HSETC
      KM(3) = K                                                        HSETC
      IF  (K.EQ.K2)  KM(1) = 0                                         HSETC
      IF  (K.EQ.K1)  KM(2) = 0                                         HSETC
      KK = IJ                                                          HSETC
      DO  12  L=1,3                                                    HSETC
      RHS(L,K) = 0.0                                                   HSETC
      IF  (KM(L).EQ.0)  GO TO 12                                       HSETC
      KK = KK+1                                                        HSETC
      DO  11  I=1,3                                                    HSETC
      DO  11  J=1,3                                                    HSETC
   11 C(I,J,KK) = 0.0                                                  HSETC
   12 CONTINUE                                                         HSETC
      IJ = IJ+1                                                        HSETC
      IJK(K,K  ) = IJ                                                  HSETC
C                                                                      HSETC
C     COMPUTE CNORM; IF ZERO, SET C(K,K) = I                           HSETC
```

```
C                                                                     HSETC
      CNORM = 0.0                                                     HSETC
      IF  (K.NE.K2)  CNORM = FB(KH)/BL(KH)                            HSETC
      IF  (K.NE.K1)  CNORM = CNORM + FB(KH-1)/BL(KH-1)                HSETC
      KI = NL(1,KNL)                                                  HSETC
      IF  (IABS(IBAR(1,KI)).GT.100)  GO TO 14                         HSETC
C     IF  (CNORM.NE.0.0)  GO TO 14                                    BUTLER1
      KK = IJK(K,K)                                                   HSETC
      DO  13  I=1,3                                                   HSETC
   13 C(I,I,KK) = ONE                                                 HSETC
      IF (CNORM.EQ.0.0) GO TO 60                                      BUTLER1
   14 KK = IBAR(3,KI)                                                 HSETC
      NFD = NTAB(KK+1)                                                HSETC
      NFR = NTAB(KK+5)                                                HSETC
C                                                                     HSETC
C     SET UP B(3,3,3) AND S(3,3)                                      HSETC
C                                                                     HSETC
      MK(1) = KH                                                      HSETC
      MK(2) = KH-1                                                    HSETC
      IF  (K.EQ.K2)  MK(1) = 0                                        HSETC
      IF  (K.EQ.K1)  MK(2) = 0                                        HSETC
      DO  18 M=1,2                                                    HSETC
      KK = MK(M)                                                      HSETC
      IF  (KK.NE.0 .AND. CNORM.NE.0.0)  GO TO 16                      HSETC
      DO  15  I=1,3                                                   HSETC
      S(I,M) = 0.0                                                    HSETC
      DO  15  J=1,3                                                   HSETC
   15 B(I,J,M) = 0.0                                                  HSETC
      GO TO 18                                                        HSETC
   16 CALL DOT31 (E(1,1,KH),U(1,KK),T)                                HSETC
      KIM = KNL + 1 - M                                               HSETC
      FB1 = FB(KK)/BL(KK)                                             HSETC
      FB2 = FP(KK)/BB(KIM) - FB1                                      HSETC
      FB3 = FP(KK)*BL(KK)/BB(KIM)**2                                  HSETC
      DO  17  I=1,3                                                   HSETC
      SGN = ONE                                                       HSETC
      IF  (FR(I,KH).LT.0.0)  SGN = -ONE                               HSETC
      S(I,M) = SGN*(FB3*T(I))                                         HSETC
      DO  17  J=1,3                                                   HSETC
   17 B(I,J,M) = SGN*(FB1*E(J,I,KH) + FB2*T(I)*U(J,KK))               HSETC
   18 CONTINUE                                                        HSETC
      DO  19  I=1,3                                                   HSETC
      S(I,3) = -(S(I,1) + S(I,2))                                     HSETC
      DO  19  J=1,3                                                   HSETC
   19 B(I,J,3) = -(B(I,J,1) + B(I,J,2))                               HSETC
      IF  (NFR.EQ.0)   GO TO 20                                       HSETC
      R(1) = TAB(NFR+2)                                               HSETC
      R(2) = TAB(NFR+4)                                               HSETC
   20 R(3) = 0.0                                                      HSETC
      DO  50  M=1,3                                                   HSETC
```

```
      RH = 0.0                                                          HSETC
      IF  (M.EQ.3)       GO TO 31                                       HSETC
      IF  (NFR.EQ.0)     GO TO 48                                       HSETC
C                                                                       HSETC
C     CONSTRAINTS 1 AND 2                                               HSETC
C                                                                       HSETC
      SGN   = -ONE                                                      HSETC
      FR3   = DABS(FR(M,KH)) - R(M)*DABS(FR(3,KH))                      HSETC
      IF  (IBAR(1,KI).GT.0)  RH = FR3                                   HSETC
      IF  (FR3.LE.0.0)  GO TO 48                                        HSETC
      GO TO 40                                                          HSETC
C                                                                       HSETC
C     CONSTRAINT NO. 3                                                  HSETC
C                                                                       HSETC
   31 IF  (NFD.EQ.0)     GO TO 48                                       HSETC
      IF  (IBAR(1,KI).LT.0)  GO TO 40                                   HSETC
      SGN   = ONE                                                       HSETC
      RMAG2 = TR(1,KH)**2 + TR(2,KH)**2 + TR(3,KH)**2                   HSETC
      RMAG  = DSQRT(RMAG2)                                              HSETC
      RER2  = TR(1,KH)*E(1,3,KH)+ TR(2,KH)*E(2,3,KH)+ TR(3,KH)*E(3,3,KH)HSETC
      RER2  = EDOT(3,KH)*RER2                                           HSETC
      RER   = DSQRT(RER2)                                               HSETC
      PEN   = RMAG/RER - RMAG                                           HSETC
      RRDOT = BAR(4,KI)*BAR(13,KI)                                      HSETC
     *        + BAR(5,KI)*BAR(14,KI)                                    HSETC
     *        + BAR(6,KI)*BAR(15,KI)                                    HSETC
      KS = IABS(IBAR(1,KI))                                             HSETC
      IF  (KS.GT.100)  KS = MOD(KS,100)                                 HSETC
      CALL DOT31 (D(1,1,KS),BAR(13,KI),T)                               HSETC
      ERDOT = E(1,3,KH)*T(1) + E(2,3,KH)*T(2) + E(3,3,KH)*T(3)          HSETC
      C1    = PEN/RMAG2                                                 HSETC
      C2    = RMAG*EDOT(3,KH)/(RER*RER2)                                HSETC
      PDOT  = C1*RRDOT - C2*ERDOT                                       HSETC
      NFDZ = IBAR(3,KI)                                                 CHGIII
      CALL FRCDFL (PEN,PDOT,NFDZ,0,FDP,ELOSS)                           CHGIII
      CALL FRCDFL (PEN,PDOT,NFDZ,1,FD ,ELOSS)                           CHGIII
      RH    = FD + FR(3,KH)                                             HSETC
      PTLOSS(2,KH) = ELOSS                                              HSETC
      C1    = FDP*C1                                                    HSETC
      C2    = FDP*C2                                                    HSETC
      SGNB3 = -DSIGN(ONE,FR(3,KH))                                      HSETC
      DO  32  J=1,3                                                     HSETC
   32 B(3,J,3) = SGNB3*B(3,J,3) - C1*TR(J,KH) + C2*E(J,3,KH)            HSETC
   40 DO  47  LL=1,3                                                    HSETC
      L = 4 - LL                                                        HSETC
      IF  (KM(L).EQ.0)  GO TO 47                                        HSETC
      DO  42  J=1,3                                                     HSETC
   42 V(J) = R(M)*B(3,J,L) + SGN*B(M,J,L)                               HSETC
      KL = KM(L)                                                        HSETC
      KML = KNL + KL - K                                                HSETC
```

```
      KIL = NL(1,KML)                                                    HSETC
      IF  (IBAR(5,KIL).NE.0)  GO TO 43                                   HSETC
      KHL = KH + KL - K                                                  HSETC
      CALL DOT31 (E(1,1,KHL),V,T)                                        HSETC
      T(2) = R(M)*S(3,L) + SGN*S(M,L)                                    HSETC
      CALL MAT31 (E(1,1,KHL),T,V)                                        HSETC
   43 IF  (LL.NE.1)  GO TO 44                                            HSETC
      VE = V(1)*E(1,M,KH) + V(2)*E(2,M,KH) + V(3)*E(3,M,KH)              HSETC
      EV = 1.0                                                           HSETC
      IF  (IABS(IBAR(1,KI)).LT.100)                                      HSETC
     *          EV = DSIGN(ONE,VE)/DSQRT(V(1)**2+V(2)**2+V(3)**2)        HSETC
      RH = EV*RH                                                         HSETC
   44 IF  (IJK(K,KL).NE.0)  GO TO 45                                     HSETC
      IJ = IJ+1                                                          HSETC
      IJK(K,KL) = IJ                                                     HSETC
   45 KK = IJK(K,KL)                                                     HSETC
      DO  46  J=1,3                                                      HSETC
      VEV = EV*V(J)                                                      HSETC
      DO  46  I=1,3                                                      HSETC
   46 C(I,J,KK) = C(I,J,KK) + E(I,M,KH)*VEV                              HSETC
   47 CONTINUE                                                           HSETC
      DO  41  I=1,3                                                      HSETC
   41 RHS(I,K) = RHS(I,K) + RH*E(I,M,KH)                                 HSETC
      GO TO 50                                                           HSETC
   48 IF  (IBAR(1,KI).LE.0)  GO TO 50                                    HSETC
      KK = IJK(K,K)                                                      HSETC
      DO  49  I=1,3                                                      HSETC
      DO  49  J=1,3                                                      HSETC
   49 C(I,J,KK) = C(I,J,KK) + E(I,M,KH)*E(J,M,KH)                        HSETC
   50 CONTINUE                                                           HSETC
   60 CONTINUE                                                           HSETC
      RETURN                                                             HSETC
      END                                                                HSETC
```

```
      SUBROUTINE HYABF(B,Z,A,F)                                         HYABF
C                                                    REV IV    02/07/87HYABF
      IMPLICIT REAL*8(A-H,O-Z)                                          HYABF
C                                                                       HYABF
C     CALCULATES A, AZ, Z.AZ:  OLD FORM MUST BE DIAGONAL                HYABF
C                                                                       HYABF
      DIMENSION B(24),Z(1),A(3,3)                                       HYABF
      P2 = 0.0                                                          HYABF
      IF(B(1).LT.0.0)P2 = -B(1) - 2.0                                   HYABF
      F = 0.0                                                           HYABF
      DO 30 I = 1,3                                                     HYABF
      J = I                                                             HYABF
      IF(B(1).LT.0.0)GO TO 10                                           HYABF
      A(I,1) = 1.0/B(I)**2                                              HYABF
      GO TO 15                                                          HYABF
   10 A(I,1) = B(I+16)                                                  HYABF
      J = J + 1                                                         HYABF
      A(I,1) = HYFCN(A(I,1),Z(I),B(J),P2)                               HYABF
C     IF(P2.GT.0.0)A(I,1) = A(I,1)*DABS(Z(I)/B(J))**P2                  HYABF
   15 DO 20 J = 2,3                                                     HYABF
   20 A(I,J) = A(I,J-1)*Z(I)                                            HYABF
   30 F = F + A(I,3)                                                    HYABF
      RETURN                                                            HYABF
      END                                                               HYABF
```

```
      SUBROUTINE HYBND(M,Z,IV,U,C,X)                                    HYBND
C                                                       REV IV    02/07/87HYBND
      IMPLICIT REAL*8(A-H,O-Z)                                          HYBND
C                                                                       HYBND
C     SEARCHES FOR POINT NEAREST CORNER - DIRECTION C*U                 HYBND
C                                                                       HYBND
      DIMENSION Z(3,12),IV(12),U(3),X(3)                                HYBND
      DO 20 I = 1,M,2                                                   HYBND
      J = IV(I)                                                         HYBND
      ATST = C*(U(1)*Z(1,J) + U(2)*Z(2,J) + U(3)*Z(3,J))                HYBND
      IF(I.EQ.1)GO TO 10                                                HYBND
      TEST = AMAX - ATST                                                HYBND
      COMP = DMAX1(DABS(AMAX),DABS(ATST))                               HYBND
C PRECISION TEST -  TRY >1000??                                         HYBND
      IF(1000.*DABS(TEST).LT.COMP)TEST = 0.0                            HYBND
      IF(TEST)10,15,20                                                  HYBND
C     IF(AMAX-ATST)10,15,20                                             HYBND
   10 AMAX = ATST                                                       HYBND
      J1 = J                                                            HYBND
   15 J2 = J                                                            HYBND
   20 CONTINUE                                                          HYBND
      DO 25 I = 1,3                                                     HYBND
   25 X(I) = 0.5*(Z(I,J1) + Z(I,J2))                                    HYBND
      RETURN                                                            HYBND
      END                                                               HYBND
```

```
      SUBROUTINE HYBOX(E,T,P,N,Z,IV)                                    HYBOX
C                                                  REV IV    02/07/87HYBOX
      IMPLICIT REAL*8(A-H,O-Z)                                          HYBOX
C                                                                       HYBOX
C     COMPUTES THE INTERSECTION OF A PLANE WITH THE EDGES OF A BOX      HYBOX
C                                                                       HYBOX
      DIMENSION T(3),E(3),TU(3),T2(3),P(3)                              HYBOX
C TO BE SAFE IV AND Z SHOULD BE DIMENSION 14 IN CALLING PROGRAM         HYBOX
      DIMENSION IV(12)                                                  HYBOX
      DIMENSION Z(3,12)                                                 HYBOX
      DATA ONE/1.0D0/                                                   HYBOX
C T - PLANE VECTOR, P POINT IN PLANE                                    HYBOX
      TUV = 0.0                                                         HYBOX
      DO 10 I = 1,3                                                     HYBOX
      TU(I) = T(I)*E(I)                                                 HYBOX
      T2(I) = 2.0*TU(I)                                                 HYBOX
   10 TUV = TUV + T(I)*(E(I) + P(I))                                    HYBOX
      N = 0                                                             HYBOX
      J = 2                                                             HYBOX
      K = 3                                                             HYBOX
      DO 45 I = 1,3                                                     HYBOX
      CK = -E(K)                                                        HYBOX
      P1 = TUV                                                          HYBOX
      DO 40 LL = 1,2                                                    HYBOX
      P2 = P1 - T2(I)                                                   HYBOX
      P3 = P2 - T2(J)                                                   HYBOX
      P4 = P1 - T2(J)                                                   HYBOX
      M = N                                                             HYBOX
      IF(DSIGN(ONE,P2).EQ.DSIGN(ONE,P1))GO TO 15                        HYBOX
      M = M + 1                                                         HYBOX
      Z(I,M) = (P1/TU(I) - 1.0)*E(I)                                    HYBOX
      Z(J,M) = -E(J)                                                    HYBOX
      Z(K,M) =  CK                                                      HYBOX
   15 IF(DSIGN(ONE,P3).EQ.DSIGN(ONE,P2))GO TO 20                        HYBOX
      M = M + 1                                                         HYBOX
      Z(I,M) =  E(I)                                                    HYBOX
      Z(J,M) = (P2/TU(J) - 1.0)*E(J)                                    HYBOX
      Z(K,M) =  CK                                                      HYBOX
   20 IF(DSIGN(ONE,P3).EQ.DSIGN(ONE,P4))GO TO 25                        HYBOX
      M = M + 1                                                         HYBOX
      Z(I,M) = (P4/TU(I) - 1.0)*E(I)                                    HYBOX
      Z(J,M) =  E(J)                                                    HYBOX
      Z(K,M) =  CK                                                      HYBOX
   25 IF(DSIGN(ONE,P4).EQ.DSIGN(ONE,P1))GO TO 30                        HYBOX
      M = M + 1                                                         HYBOX
      Z(I,M) = -E(I)                                                    HYBOX
      Z(J,M) = (P1/TU(J) - 1.0)*E(J)                                    HYBOX
      Z(K,M) =  CK                                                      HYBOX
   30 IF(M.EQ.N)GO TO 35                                                HYBOX
CHECK FOR PRECISION (+-+-.OR -+-+)                                      HYBOX
```

```
      IF(M.EQ.N+4)GO TO 35                                              HYBOX
C DELETE 0 LENGTH SIDE                                                  HYBOX
      IF((Z(I,M-1).EQ.Z(I,M)).AND.(Z(J,M-1).EQ.Z(J,M)))GO TO 35         HYBOX
      N = M                                                             HYBOX
   35 P1 = P1 - T2(K)                                                   HYBOX
   40 CK = -CK                                                          HYBOX
      J = K                                                             HYBOX
   45 K = I                                                             HYBOX
C                                                                       HYBOX
      IF(N.LT.6)GO TO 65                                                HYBOX
      IV(1) = 1                                                         HYBOX
      IV(2) = 2                                                         HYBOX
      M = 2                                                             HYBOX
      DO 60 J = 3,N,2                                                   HYBOX
      D = DABS(Z(1,M)) + DABS(Z(2,M)) + DABS(Z(3,M))                    HYBOX
      DO 55 L = 3,N                                                     HYBOX
      DO 50 LL = 2,J                                                    HYBOX
      IF(IV(LL-1).EQ.L) GO TO 55                                        HYBOX
   50 CONTINUE                                                          HYBOX
      F = DABS(Z(1,M)-Z(1,L))+DABS(Z(2,M)-Z(2,L))+DABS(Z(3,M)-Z(3,L))   HYBOX
      IF(F.GT.D)GO TO 55                                                HYBOX
      D = F                                                             HYBOX
      K = L                                                             HYBOX
   55 CONTINUE                                                          HYBOX
      M = K + 1                                                         HYBOX
      IF(MOD(K,2).EQ.0)M = K - 1                                        HYBOX
      IV(J) = K                                                         HYBOX
      IV(J+1) = M                                                       HYBOX
   60 CONTINUE                                                          HYBOX
   65 RETURN                                                            HYBOX
      END                                                               HYBOX
```

```
      SUBROUTINE HYDAD(D,A,DAD)                                         HYDAD
C                                                REV IV    02/07/87HYDAD
      IMPLICIT REAL*8(A-H,O-Z)                                          HYDAD
COMPUTES D'A(*,1)D                                                      HYDAD
      DIMENSION D(3,3),A(3),DAD(3,3)                                    HYDAD
      DO 10 I = 1,3                                                     HYDAD
      DO 10 J = 1,3                                                     HYDAD
      DAD(I,J) = 0.0                                                    HYDAD
      DO 10 K = 1,3                                                     HYDAD
   10 DAD(I,J) = DAD(I,J) + D(K,I)*A(K)*D(K,J)                          HYDAD
      RETURN                                                            HYDAD
      END                                                               HYDAD
```

```
      SUBROUTINE HYEST(BM,BN,TAB)                                       HYEST
C                                                    REV IV    07/23/87HYEST
C LINEAR PROGRAM                                                        HYEST
      IMPLICIT REAL*8(A-H,O-Z)                                          HYEST
      DIMENSION BM(24),BN(24),TAB(8)                                    HYEST
      COMMON/TEMPVS/D12(3,3),A(3,3),B(3,3),XMN(3),RLN(3),XMM(3),        HYEST
     *              T(3),R(3),C(3,3),V(7)                               HYEST
C R GOES FROM M TO N   D12 = DM*DN'                                     HYEST
C R = 0 CANNOT BE SOLVED WITH THIS METHOD                               HYEST
      BE = 1.0                                                          HYEST
      RR = R(1)**2 + R(2)**2 + R(3)**2                                  HYEST
      IF(RR.EQ.0.0)GO TO 30                                             HYEST
C R.R = 0 INVALID                                                       HYEST
      M = 1                                                             HYEST
      N = 1                                                             HYEST
      IF(BM(1).LT.0.0)M = 2                                             HYEST
      IF(BN(1).LT.0.0)N = 2                                             HYEST
      PM = 2.                                                           HYEST
      PN = 2.                                                           HYEST
      IF(M.EQ.2)PM = -BM(1)                                             HYEST
      IF(N.EQ.2)PN = -BN(1)                                             HYEST
      DO 10 I = 1,3                                                     HYEST
      T(I) = R(I)                                                       HYEST
      DO 10 J = 1,3                                                     HYEST
   10 B(I,J) = D12(I,J)                                                 HYEST
      IF(N.EQ.2)CALL DOTT33(D12,BN(8),B)                                HYEST
      DO 15 I = 1,3                                                     HYEST
      DO 15 J = 1,3                                                     HYEST
   15 C(I,J) = B(I,J)                                                   HYEST
      IF(M.EQ.2)CALL MAT33(BM(8),B,C)                                   HYEST
C C WILL TRANSFORM FROM NN TO MM                                        HYEST
      IF(M.EQ.2)CALL MAT31(BM(8),R,T)                                   HYEST
      CALL HYLPX(BM(M),BN(N))                                           HYEST
      BE = V(7)                                                         HYEST
      IF(V(7).LE.1.0)GO TO 30                                           HYEST
      CALL HYABF(BM(1),V(1),A,F1)                                       HYEST
      CALL HYABF(BN(1),V(4),B,F2)                                       HYEST
C ESTIMATE ALPHA                                                        HYEST
      AA = A(1,2)**2 + A(2,2)**2 + A(3,2)**2                            HYEST
      BB = B(1,2)**2 + B(2,2)**2 + B(3,2)**2                            HYEST
      ALP = DSQRT(AA/BB)                                                HYEST
      RA = F1**(1.0/PM)                                                 HYEST
      RB = F2**(1.0/PN)                                                 HYEST
      ALP = ALP*RA*F2/(RB*F1)                                           HYEST
C SCALE POINTS TO ELLIPSOIDS                                            HYEST
      DO 20 I = 1,3                                                     HYEST
      V(I)    = V(I)/RA                                                 HYEST
   20 V(I+3) = V(I+3)/RB                                                HYEST
C ESTIMATE BETA                                                         HYEST
      CALL MAT31(C,V(4),T)                                              HYEST
```

```
      BE = (V(1)-T(1))**2 + (V(2)-T(2))**2 + (V(3) - T(3))**2              HYEST
      BE = DSQRT(BE/RR)                                                     HYEST
C STORE VALUES IN TAB ARRAY FOR CONTACT                                     HYEST
      TAB(1) = ALP                                                          HYEST
      DO 25 I = 1,6                                                         HYEST
   25 TAB(I+2) = V(I)                                                       HYEST
   30 TAB(2) = BE                                                           HYEST
      RETURN                                                                HYEST
      END                                                                   HYEST
```

```
      DOUBLE PRECISION FUNCTION HYFCN(C,Z,A,P)                          HYFCN
C                                                    REV IV    02/07/87HYFCN
      IMPLICIT REAL*8(A-H,O-Z)                                          HYFCN
      HYFCN = C                                                         HYFCN
      IF(P.EQ.0.0)GO TO 10                                              HYFCN
      HYFCN = 0.0                                                       HYFCN
      IF(Z.EQ.0.0)GO TO 10                                              HYFCN
      Q = P*(DLOG(DABS(Z)) - DLOG(A))                                   HYFCN
      IF(Q.GT.-88.5) HYFCN = C*DEXP(Q)                                  HYFCN
   10 RETURN                                                            HYFCN
      END                                                               HYFCN
```

```
      SUBROUTINE HYLIM(A,U,B,V,C,W,Z,BD)                                HYLIM
C                                                REV IV    12/11/87HYFIX
      IMPLICIT REAL*8(A-H,O-Z)                                          HYLIM
C GIVEN Z, FIND A,B,Z: ZEZ = 1,  ZEV = 0,  TZ = TP                      HYLIM
      DIMENSION BD(24)                                                  HYLIM
      DIMENSION U(3),V(3),W(3),EI(3),EJ(3),T(3),TV(3)                   HYLIM
      DIMENSION Z(3),E(3),EV(3),Q(3),S(3),EZ(3)                         HYLIM
      DIMENSION SM(3,3)                                                 HYLIM
      LOGICAL PASS,USEV                                                 HYLIM
      PASS = .FALSE.                                                    HYLIM
      ITER = 100                                                        HYLIM
      PP =  -1./BD(1)                                                   HYLIM
      POW = -BD(1) - 2.0                                                HYLIM
      P1  = -BD(1) - 1.0                                                HYLIM
      P01 =  1.0/P1                                                     HYLIM
      P2  =  -BD(1)/P1                                                  HYLIM
      DO 10 I = 1,3                                                     HYLIM
      TV(I) = 0.0                                                       HYLIM
   10 IF(V(I).NE.0.0)TV(I) = HYFCN(1.0/V(I),V(I),BD(I+1),P2)            HYFIX
C GET RECIPROCAL SET                                                    HYLIM
      CALL CROSS(V,W,EI)                                                HYLIM
      CALL CROSS(W,U,EJ)                                                HYLIM
      CALL CROSS(U,V,T)                                                 HYLIM
      EIU = EI(1)*U(1) + EI(2)*U(2) + EI(3)*U(3)                        HYLIM
      G = C*EIU                                                         HYLIM
C                                                                       HYLIM
      DO 55 IT = 1,ITER                                                 HYLIM
      EVM = 0.0                                                         HYLIM
      EVZ = 0.0                                                         HYLIM
      ZEZ = 0.0                                                         HYLIM
      USEV = .FALSE.                                                    HYLIM
      DO 15 I = 1,3                                                     HYLIM
      E(I) = HYFCN(BD(I+16),Z(I),BD(I+1),POW)                           HYLIM
      EV(I) = E(I)*V(I)                                                 HYLIM
      IF(EV(I).EQ.0.0)USEV = .TRUE.                                     HYLIM
      IF(DABS(EV(I)).GT.EVM)EVM = DABS(EV(I))                           HYLIM
      EZ(I) = E(I)*Z(I)                                                 HYLIM
      IF(DABS(EZ(I)).GT.EVZ)EVZ = DABS(EZ(I))                           HYLIM
   15 ZEZ = ZEZ + Z(I)*EZ(I)                                            HYLIM
      RHO  = ZEZ**PP                                                    HYLIM
      DO 20 I = 1,3                                                     HYLIM
   20 Z(I) = Z(I)/RHO                                                   HYLIM
      IF(PASS)GO TO 60                                                  HYLIM
      RHOZ = ZEZ/RHO                                                    HYLIM
      RHOV = EVM*RHOZ/RHO                                               HYLIM
      RHOZ = EVZ*RHOZ                                                   HYLIM
      IF(.NOT.USEV)GO TO 30                                             HYLIM
      RHOV = 1.0                                                        HYLIM
      DO 25 I = 1,3                                                     HYLIM
   25 EV(I) = TV(I)                                                     HYLIM
```

```
C WHAT IF NO TV IS 0 AND EV ARE ALL 0 ?                                HYLIM
   30 DO 35 I = 1,3                                                    HYLIM
      EV(I) = EV(I)/RHOV                                               HYLIM
   35 EZ(I) = EZ(I)/RHOZ                                               HYLIM
C SET UP MATRIX                                                        HYLIM
      CALL CROSS(EV, T,SM(1,1))                                        HYLIM
      CALL CROSS(T ,EZ,SM(1,2))                                        HYLIM
      CALL CROSS(EZ,EV,SM(1,3))                                        HYLIM
      TZV = T(1)*SM(1,3) + T(2)*SM(2,3) + T(3)*SM(3,3)                 HYLIM
      TZ  = T(1)*Z(1)    + T(2)*Z(2)    + T(3)*Z(3)                    HYLIM
      ZEV = Z(1)*EV(1)   + Z(2)*EV(2)   + Z(3)*EV(3)                   HYLIM
      IF(TZV.EQ.0.0)STOP 39                                            HYLIM
      ZEV = ZEV/TZV                                                    HYLIM
      Q(1) = 0.0                                                       HYLIM
      Q(2) = -ZEV                                                      HYLIM
      IF(.NOT.USEV)Q(2) = Q(2)/P1                                      HYLIM
      Q(3) = (G - TZ)/TZV                                              HYLIM
      CALL MAT31(SM,Q,S)                                               HYLIM
      SS = 0.0                                                         HYLIM
      ZZ = 0.0                                                         HYLIM
      DO 50 I = 1,3                                                    HYLIM
      SS = SS + DABS(S(I))                                             HYLIM
      IF(DABS(Z(I)).LT.0.1*BD(I+1))GO TO 45                            HYLIM
      IF(DABS(S(I)).GT.DABS(Z(I)))S(I) = DSIGN(0.5*Z(I),S(I))          HYLIM
   45 Z(I) = Z(I) + S(I)                                               HYLIM
      IF(DABS(Z(I)).GT.BD(I+1))Z(I) = DSIGN(BD(I+1),Z(I))              HYLIM
   50 ZZ = ZZ + DABS(Z(I))                                             HYLIM
      IF(SS.LT.1.0E-10*ZZ)PASS = .TRUE.                                HYLIM
   55 CONTINUE                                                         HYLIM
C                                                                      HYLIM
   60 A = (EI(1)*Z(1) + EI(2)*Z(2) + EI(3)*Z(3))/EIU                   HYLIM
      B = (EJ(1)*Z(1) + EJ(2)*Z(2) + EJ(3)*Z(3))/EIU                   HYLIM
      RETURN                                                           HYLIM
      END                                                              HYLIM
```

```
      SUBROUTINE HYLPR(J1,J2,ID,C,S,E,T)                                HYLPR
C                                                    REV IV    02/07/87HYLPR
      IMPLICIT REAL*8(A-H,O-Z)                                          HYLPR
      DIMENSION ID(16),C(16),S(9,8),E(9),T(7)                           HYLPR
C LINEAR PROGRAM ROUTINE USING SIMPLEX METHOD                           HYLPR
C J1 = J2 FORCED PIVOT ON COLUMN J1                                     HYLPR
CALCULATE COSTS                                                         HYLPR
      J = J1                                                            HYLPR
      IF(J.EQ.J2)GO TO 30                                               HYLPR
   10 DO 20 L = 1,7                                                     HYLPR
      T(L) = -C(L)                                                      HYLPR
      IF(C(L).EQ.10.)GO TO 20                                           HYLPR
      DO 15 I = 1,9                                                     HYLPR
   15 T(L) = T(L) + S(I,L)*C(I+7)                                       HYLPR
   20 CONTINUE                                                          HYLPR
      IF(J1.EQ.J2)GO TO 65                                              HYLPR
C FIND PIVOT COLUMN                                                     HYLPR
      DO 25 L = 1,7                                                     HYLPR
      J = L                                                             HYLPR
      IF(T(L).GT.0.0)GO TO 30                                           HYLPR
   25 CONTINUE                                                          HYLPR
      GO TO 65                                                          HYLPR
C FIND PIVOT ROW                                                        HYLPR
   30 K = 0                                                             HYLPR
      DO 40 I = 1,9                                                     HYLPR
C SAVE PIVOT COLUMN                                                     HYLPR
      E(I) = S(I,J)                                                     HYLPR
      IF(S(I,J).LE.0.0)GO TO 40                                         HYLPR
      IF(K.EQ.0)GO TO 35                                                HYLPR
      IF(S(I,8).GE.Z*S(I,J))GO TO 40                                    HYLPR
   35 K = I                                                             HYLPR
      Z = S(I,8)/S(I,J)                                                 HYLPR
   40 CONTINUE                                                          HYLPR
C REPLACE COLUMNS                                                       HYLPR
      IF(K.EQ.0)GO TO 65                                                HYLPR
      M = ID(J)                                                         HYLPR
      ID(J) = ID(K+7)                                                   HYLPR
      ID(K+7) = M                                                       HYLPR
      Q = C(J)                                                          HYLPR
      C(J) = C(K+7)                                                     HYLPR
      C(K+7) = Q                                                        HYLPR
      P = S(K,J)                                                        HYLPR
      DO 45 I = 1,9                                                     HYLPR
   45 S(I,J) = 0.0                                                      HYLPR
      S(K,J) = 1.0                                                      HYLPR
      DO 50 L = 1,8                                                     HYLPR
   50 S(K,L) = S(K,L)/P                                                 HYLPR
      E(K) = 1.0                                                        HYLPR
      DO 60 I = 1,9                                                     HYLPR
      IF(I.EQ.K)GO TO 60                                                HYLPR
```

```
      IF(E(I).EQ.0.0)GO TO 60                                                    HYLPR
      DO 55 M = 1,8                                                              HYLPR
   55 S(I,M) = S(I,M) - E(I)*S(K,M)                                              HYLPR
   60 CONTINUE                                                                   HYLPR
      GO TO 10                                                                   HYLPR
   65 RETURN                                                                     HYLPR
      END                                                                        HYLPR
```

```
      SUBROUTINE HYLPX(BM,BN)                                           HYLPX
C                                                  REV IV    02/07/87HYLPX
C LINEAR PROGRAM EXEC                                                  HYLPX
      IMPLICIT REAL*8(A-H,O-Z)                                          HYLPX
      DIMENSION BM(23),BN(23)                                           HYLPX
      COMMON/TEMPVS/D12(3,3),P(3,3),Q(3,3),XMN(3),RLN(3),XMM(3),        HYLPX
     *              R(3),H(3),D(3,3),V(7),S(9,8),C(16),A(7),B(3),       HYLPX
     *              E(9),T(7),ID(16),IP(2)                              HYLPX
      CALL MAT31(D,BN,B)                                                HYLPX
      DO 10 I = 1,3                                                     HYLPX
      B(I) = BM(I) - B(I) + R(I)                                        HYLPX
      A(I) = BM(I)                                                      HYLPX
   10 A(I+3) = BN(I)                                                    HYLPX
      A(7) = -1.0                                                       HYLPX
      DO 15 I = 1,16                                                    HYLPX
      C(I) = 0.0                                                        HYLPX
   15 ID(I) = I                                                         HYLPX
      C(7) = -1.0                                                       HYLPX
      DO 20 I = 1,9                                                     HYLPX
      DO 20 J = 1,8                                                     HYLPX
   20 S(I,J) = 0.0                                                      HYLPX
C                                                                       HYLPX
COSTS 0    0   -1                                                       HYLPX
C     I   -D   -R    A - DB  + R  (>0)      INF COST                    HYLPX
C     I    0    0    2A                                                 HYLPX
C     0    I    0    2B                                                 HYLPX
C                                                                       HYLPX
      DO 25 I = 1,6                                                     HYLPX
   25 S(I+3,I) = 1.0                                                    HYLPX
      DO 30 I = 1,3                                                     HYLPX
      C(I+7) = 10.                                                      HYLPX
      S(I,7) = -R(I)                                                    HYLPX
      S(I,I) = 1.0                                                      HYLPX
      S(I,8) = B(I)                                                     HYLPX
      S(I+3,8) = 2.0*A(I)                                               HYLPX
      S(I+6,8) = 2.0*A(I+3)                                             HYLPX
      DO 30 J = 1,3                                                     HYLPX
   30 S(I,J+3) = -D(I,J)                                                HYLPX
CHECK SIGN OF RHS                                                       HYLPX
      DO 40 I = 1,3                                                     HYLPX
      IF(B(I).GE.0.0)GO TO 40                                           HYLPX
      DO 35 J = 1,8                                                     HYLPX
   35 S(I,J) = -S(I,J)                                                  HYLPX
   40 CONTINUE                                                          HYLPX
      J1 = 1                                                            HYLPX
      J2 = 7                                                            HYLPX
C                                                                       HYLPX
      CALL HYLPR(J1,J2,ID,C,S,E,T)                                      HYLPX
      NK = 0                                                            HYLPX
      NZ = 0                                                            HYLPX
```

```
COUNT ZEROES IN SOLUTION                                          HYLPX
      DO 45 I = 1,7                                               HYLPX
C TEST SHOULD PROBABLY BE AN EPSILON TEST, DABS(T(I)).GT.EPS      HYLPX
      IF((T(I).NE.0.0).OR.(C(I).EQ.10.))GO TO 45                  HYLPX
      NZ = NZ + 1                                                 HYLPX
      IP(NZ) = I                                                  HYLPX
   45 CONTINUE                                                    HYLPX
C SET PASS COUNT                                                  HYLPX
      NP = 1                                                      HYLPX
      IF(NZ.GT.0)NP = 2**NZ                                       HYLPX
C                                                                 HYLPX
      DO 55 M = 1,NP                                              HYLPX
      NM = NK                                                     HYLPX
      NK = NK + 1                                                 HYLPX
      DO 50 I = 1,16                                              HYLPX
      K = ID(I)                                                   HYLPX
      IF(K.GT.7)GO TO 50                                          HYLPX
      W = 0.0                                                     HYLPX
      IF(I.GT.7)W = S(I-7,8)                                      HYLPX
      V(K) = (W - A(K) + NM*V(K))/NK                              HYLPX
   50 CONTINUE                                                    HYLPX
C LOOK FOR ALL SOLUTIONS                                          HYLPX
      IF(M.EQ.NP)GO TO 55                                         HYLPX
      J1 = IP(1)                                                  HYLPX
      IP(1) = IP(2)                                               HYLPX
      IP(2) = J1                                                  HYLPX
      IF(T(J1).NE.0.0)GO TO 55                                    HYLPX
      J2 = J1                                                     HYLPX
      CALL HYLPR(J1,J2,ID,C,S,E,T)                                HYLPX
   55 CONTINUE                                                    HYLPX
      RETURN                                                      HYLPX
      END                                                         HYLPX
```

```
      SUBROUTINE HYNTR(BM,BN,TAB)                                       HYNTR
C                                                 REV IV    02/07/87HYNTR
      IMPLICIT REAL*8(A-H,O-Z)                                          HYNTR
CALCULATIONS IN SEGMENT M'S REFERENCE                                   HYNTR
      DIMENSION BM(24),BN(24),TAB(8)                                    HYNTR
      COMMON/TEMPVS/D12(3,3),A(3,3),B(3,3),XMN(3),RLN(3),XMM(3),        HYNTR
     *             AZ(3),R(3),Z(3),DNM(3,3),DAD(3,3),DBD(3,3),          HYNTR
     *             BMD(3,3),TMP(3,3),S(5,6),ZVR(3),BZV(3),V(3),         HYNTR
     *             ZM(3),VN(3),F(2),BV(3)                               HYNTR
      P1 = 2.0                                                          HYNTR
      P2 = 2.0                                                          HYNTR
      IF(BM(1).LT.-2.0)P1 = -BM(1)                                      HYNTR
      IF(BN(1).LT.-2.0)P2 = -BN(1)                                      HYNTR
      C1 =  P1 - 1.0                                                    HYNTR
      CN =  P2 - 1.0                                                    HYNTR
C DNM TRANSFORMS FROM M TO NN                                           HYNTR
      K = 8                                                             HYNTR
      DO 15 J = 1,3                                                     HYNTR
      DO 10 I = 1,3                                                     HYNTR
      BMD(I,J) = 0.0                                                    HYNTR
      IF(BM(1).LT.0.0)BMD(I,J) = BM(K)                                  HYNTR
      DNM(I,J) = D12(I,J)                                               HYNTR
   10 K = K + 1                                                         HYNTR
   15 IF(BM(1).GT.0.0)BMD(J,J) = 1.0                                    HYNTR
      IF(BN(1).LT.0.0)CALL DOTT33(BN(8),D12,DNM)                        HYNTR
      ALP = TAB(1)                                                      HYNTR
      BET = TAB(2)                                                      HYNTR
      DO 20 I = 1,3                                                     HYNTR
      ZM(I) = TAB(I+2)                                                  HYNTR
   20 VN(I) = TAB(I+5)                                                  HYNTR
C PUT VECTORS INTO N'S REFERENCE                                        HYNTR
      CALL DOT31(BMD,ZM,Z)                                              HYNTR
      CALL DOT31(DNM,VN,V)                                              HYNTR
      DO 25 I = 1,3                                                     HYNTR
   25 ZVR(I) = Z(I) - V(I) - BET*R(I)                                   HYNTR
C                                                                       HYNTR
      DO 40 ITER = 1,100                                                HYNTR
      CALL HYABF(BM,ZM,A,F(1))                                          HYNTR
      CALL HYABF(BN,VN,B,F(2))                                          HYNTR
      CALL DOT31(BMD,A(1,2),AZ)                                         HYNTR
      CALL DOT31(DNM,B(1,2),BV)                                         HYNTR
      CALL HYDAD(BMD,A,DAD)                                             HYNTR
      CALL HYDAD(DNM,B,DBD)                                             HYNTR
      CALL MAT31(DBD,R,S(1,5))                                          HYNTR
      CALL MAT31(DBD,ZVR,BZV)                                           HYNTR
      C2 = CN*ALP                                                       HYNTR
C SET UP S MATRIX                                                       HYNTR
      S(4,4) = 0.0                                                      HYNTR
      S(5,5) = 0.0                                                      HYNTR
      S(4,6) = (1.0 - F(2))/P2 -V(1)*BZV(1) -V(2)*BZV(2) -V(3)*BZV(3)   HYNTR
```

```
      S(5,6) = (1.0 - F(1))/P1                                          HYNTR
      S(4,5) = -BV(1)*R(1) - BV(2)*R(2) - BV(3)*R(3)                    HYNTR
      S(5,4) = 0.0                                                      HYNTR
      DO 30 I = 1,3                                                     HYNTR
      S(I,4) =  BV(I)                                                   HYNTR
      S(4,I) =  BV(I)                                                   HYNTR
      S(I,5) = -C2*S(I,5)                                               HYNTR
      S(5,I) =  AZ(I)                                                   HYNTR
      S(I,6) = -AZ(I) - ALP*BV(I) - C2*BZV(I)                           HYNTR
      DO 30 J = 1,3                                                     HYNTR
   30 S(I,J) = C1*DAD(I,J) + C2*DBD(I,J)                                HYNTR
      CALL HYSOL(S,5,5)                                                 HYNTR
      TALP = ALP + S(4,6)                                               HYNTR
      IF(TALP.LE.0.0)TALP = ALP/2.0                                     HYNTR
      ALP = TALP                                                        HYNTR
      TBET = BET + S(5,6)                                               HYNTR
      IF(TBET.LE.0.0)TBET = BET/2.0                                     HYNTR
      BET = TBET                                                        HYNTR
      SS = 0.0                                                          HYNTR
      ZZ = 0.0                                                          HYNTR
      DO 35 I = 1,3                                                     HYNTR
      SS   = SS + DABS(S(I,6))                                          HYNTR
      Z(I) = Z(I) + S(I,6)                                              HYNTR
      ZZ   = ZZ + DABS(Z(I))                                            HYNTR
      V(I) = Z(I) - BET*R(I)                                            HYNTR
   35 ZVR(I) = 0.0                                                      HYNTR
      CALL MAT31(BMD,Z,ZM)                                              HYNTR
      CALL MAT31(DNM,V,VN)                                              HYNTR
CONVERGENCE TEST DEPENDS ON REAL*4 (1.0E-5) OR REAL*8 (??)              HYNTR
      IF(SS.LT.1.0E-10*ZZ)GO TO 50                                      HYNTR
      K = 1                                                             HYNTR
      L = 1                                                             HYNTR
      IF(BM(1).LT.0.0)K = 2                                             HYNTR
      IF(BN(1).LT.0.0)L = 2                                             HYNTR
      DO 37 I = 1,3                                                     HYNTR
      IF(DABS(ZM(I)).GT.BM(K)) ZM(I) = DSIGN(BM(K),ZM(I))               HYNTR
      IF(DABS(VN(I)).GT.BN(L)) VN(I) = DSIGN(BN(L),VN(I))               HYNTR
      K = K + 1                                                         HYNTR
   37 L = L + 1                                                         HYNTR
      CALL DOT31(BMD,ZM,Z)                                              HYNTR
      CALL DOT31(DNM,VN,V)                                              HYNTR
      DO 38 I = 1,3                                                     HYNTR
   38 ZVR(I) = Z(I) - V(I) - BET*R(I)                                   HYNTR
   40 CONTINUE                                                          HYNTR
C     WRITE(6,45)                                                       HYNTR
C  45 FORMAT(' HYNTR DID NOT CONVERGE, CONTACT IGNORED.')               HYNTR
      BET = 1.0                                                         HYNTR
   50 TAB(1) = ALP                                                      HYNTR
      TAB(2) = BET                                                      HYNTR
      DO 55 I = 1,3                                                     HYNTR
```

```
      TAB(I+2) = ZM(I)                                              HYNTR
   55 TAB(I+5) = VN(I)                                              HYNTR
      RETURN                                                        HYNTR
      END                                                           HYNTR
```

```
      DOUBLE PRECISION FUNCTION HYPEN(BD,E,V)                           HYPEN
C                                                 REV IV    02/07/87HYPEN
C  POINT OF MAXIMUM PENETRATION                                         HYPEN
C  SOLVES FOR VALUE OF ALP USED BY PLELP                                HYPEN
C  POWERS OF HYPERELLIPSOID MAY BE DIFFERENT                            HYPEN
      IMPLICIT REAL*8(A-H,O-Z)                                          HYPEN
      DIMENSION BD(24),E(3),V(3)                                        HYPEN
      FX(A) = A**E(1)*V(1)+A**E(2)*V(2)+A**E(3)*V(3)-1.0                HYPEN
      L = 1                                                             HYPEN
      VM = V(1)                                                         HYPEN
      DO 10 I = 2,3                                                     HYPEN
      IF (V(I).LE.VM) GO TO 10                                          HYPEN
      L = I                                                             HYPEN
      VM = V(I)                                                         HYPEN
   10 CONTINUE                                                          HYPEN
      A = V(1) + V(2) + V(3)                                            HYPEN
      A = 1.0/A**(1.0/E(L))                                             HYPEN
      DEL = A/2.0                                                       HYPEN
      AP = 0.0                                                          HYPEN
   12 F = FX(A)                                                         HYPEN
      IF (DABS(F).LT.1.D-08) GO TO 40                                   HYPEN
      IF (F) 16,40,14                                                   HYPEN
   14 IF (A-DEL.LE.0.0) DEL = A/2.0                                     HYPEN
      AP = A                                                            HYPEN
      FP = F                                                            HYPEN
      A = A - DEL                                                       HYPEN
      GO TO 12                                                          HYPEN
   16 IF (AP.NE.0.0) GO TO 18                                           HYPEN
      A = A + DEL                                                       HYPEN
      GO TO 12                                                          HYPEN
   18 AM = A                                                            HYPEN
      FM = F                                                            HYPEN
   20 IF (FP.EQ.FM) GO TO 40                                            HYPEN
      DEL = -FM*(AP - AM)/(FP - FM)                                     HYPEN
      AN = AM + DEL                                                     HYPEN
      IF (AN.EQ.A) GO TO 40                                             HYPEN
      A = AN                                                            HYPEN
      F = FX(A)                                                         HYPEN
      IF (DABS(F).LT.1.D-08) GO TO 40                                   HYPEN
      IF (F) 18,40,22                                                   HYPEN
   22 FP = F                                                            HYPEN
      AP = A                                                            HYPEN
      GO TO 20                                                          HYPEN
   40 HYPEN = A                                                         HYPEN
      RETURN                                                            HYPEN
      END                                                               HYPEN
```

```
      SUBROUTINE HYREA (L,H,AREA,AB,BB)                                HYFIX
C                                                    REV IV    12/11/87HYFIX
      IMPLICIT REAL*8(A-H,O-Z)                                         HYREA
      DIMENSION H(2,2,5)                                               HYFIX
      AREA = 0.0                                                       HYREA
      AB = 0.0                                                         HYREA
      BB = 0.0                                                         HYREA
      IF (L.LT.2) GO TO 20                                             HYFIX
      DO 15 I = 1,L                                                    HYFIX
      AR = H(1,1,I)*H(2,2,I) - H(1,2,I)*H(2,1,I)                       HYFIX
      IF (AR.EQ.0.0) GO TO 5                                           HYFIX
      AB = AB + AR*(H(1,1,I) + H(1,2,I))                               HYFIX
      BB = BB + AR*(H(2,1,I) + H(2,2,I))                               HYFIX
      AREA = AREA + AR                                                 HYREA
    5 AR = H(1,2,I)*H(2,1,I+1) - H(1,1,I+1)*H(2,2,I)                   HYFIX
      IF (AR.EQ.0.0) GO TO 15                                          HYFIX
      AB = AB + AR*(H(1,1,I+1) + H(1,2,I))                             HYFIX
      BB = BB + AR*(H(2,1,I+1) + H(2,2,I))                             HYFIX
      AREA = AREA + AR                                                 HYREA
   15 CONTINUE                                                         HYFIX
      IF (AREA.LE.0.0) GO TO 20                                        HYFIX
      AREA = 3.0*AREA                                                  HYREA
      AB = AB/AREA                                                     HYREA
      BB = BB/AREA                                                     HYREA
C     AREA = AREA/6.0                                                  HYREA
   20 RETURN                                                           HYREA
      END                                                              HYREA
```

```
      SUBROUTINE HYSOL(A,N,ND)                                            HYSOL
C                                                       REV IV    02/01/88MISDOT
      IMPLICIT REAL*8(A-H,O-Z)                                            HYSOL
      DIMENSION A(ND,6)                                                   MISDOT
C ASSUMES PIVOT ON DIAGONAL , BYPASS 0'S                                  HYSOL
      N1 = N + 1                                                          HYSOL
      DO 20 L = 1,N                                                       HYSOL
      IF(A(L,L).EQ.0.0)GO TO 20                                           HYSOL
      L1 = L + 1                                                          HYSOL
      DO 10 J = L1,N1                                                     HYSOL
   10 A(L,J) = A(L,J)/A(L,L)                                              HYSOL
      IF(L.EQ.N)GO TO 20                                                  HYSOL
      DO 21 I = L1,N                                                      HYSOL
      IF(A(I,L).EQ.0.0)GO TO 21                                           HYSOL
      DO 15 J = L1,N1                                                     HYSOL
   15 A(I,J) = A(I,J) - A(I,L)*A(L,J)                                     HYSOL
   21 CONTINUE                                                            HYSOL
   20 CONTINUE                                                            HYSOL
      IF(N.EQ.1)GO TO 30                                                  HYSOL
C BACKUP                                                                  HYSOL
      DO 25 L = 2,N                                                       HYSOL
      I = N1 - L                                                          HYSOL
      L1 = I + 1                                                          HYSOL
      DO 25 J = L1,N                                                      HYSOL
   25 A(I,N1) = A(I,N1) - A(I,J)*A(J,N1)                                  HYSOL
   30 RETURN                                                              HYSOL
      END                                                                 HYSOL
```

```
      SUBROUTINE HYVAL(A,U,R,BD,L)                                      HYVAL
C                                                     REV IV    12/11/87HYFIX
      IMPLICIT REAL*8(A-H,O-Z)                                          HYVAL
C GIVEN A,U,R: COMPUTE A        Z = A*U + R                             HYVAL
      DIMENSION BD(24),U(3),R(3),RM(2)                                  HYFIX
      ONE = 1.0                                                         HYFIX
      POW = -BD(1) - 2.0                                                HYVAL
C ARE THESE THE CORRECT TESTS??                                         HYFIX
      TEST = -BD(1)*0.000001                                            HYFIX
      TESD = 0.000001                                                   HYFIX
      CALL HYVBX(U,R,BD(2),M,RM)                                        HYVAL
      A = 0.0                                                           HYFIX
      IF (M.LT.L) GO TO 50                                              HYFIX
C THIS SHOULD NEVER HAPPEN  - IMPLIES R IS OUTSIDE BOX                  HYFIX
      A = RM(L)                                                         HYVAL
      IF (DABS(A).LT.TESD) GO TO 50                                     HYFIX
      DEL = A/5.0                                                       HYFIX
      NSTEP = 0                                                         HYFIX
C ITERATION LOOP                                                        HYFIX
   10 DEL = DEL/4.0                                                     HYFIX
      NSTEP = NSTEP + 1                                                 HYFIX
      IF (NSTEP.LT.100) GO TO 12                                        HYFIX
      WRITE(6,11) M,A,DEL,F1,F2,L,RM(1),RM(2),U,R,BD                    HYFIX
   11 FORMAT(' HYV ',I4,4F11.6,I3,2F11.6/4X,3F11.6,4X,3F11.6/           HYFIX
     *   4(2X,7F10.4/))                                                 HYFIX
      STOP 102                                                          HYFIX
   12 F2 = HYVFN(A,U,R,BD,POW)                                          HYFIX
      IF (DABS(F2).LT.TEST) GO TO 50                                    HYFIX
      IF (F2) 20,50,30                                                  HYFIX
   15 F2 = HYVFN(A,U,R,BD,POW)                                          HYFIX
      NSTEP = NSTEP + 1                                                 HYFIX
      IF (NSTEP.LT.100) GO TO 17                                        HYFIX
      WRITE(6,11) M,A,DEL,F1,F2,L,RM(1),RM(2),U,R,BD                    HYFIX
      STOP 103                                                          HYFIX
   17 IF (DABS(F2).LT.TEST) GO TO 50                                    HYFIX
      IF (F2) 20,50,35                                                  HYFIX
   20 IF (DSIGN(ONE,A).EQ.DSIGN(ONE,A+DEL)) GO TO 22                    HYFIX
      A = A/2.0                                                         HYFIX
      DL = -A                                                           HYFIX
      GO TO 23                                                          HYFIX
   22 DL = DEL                                                          HYFIX
      A = A + DEL                                                       HYFIX
   23 F1 = F2                                                           HYFIX
      GO TO 15                                                          HYFIX
   25 F2 = HYVFN(A,U,R,BD,POW)                                          HYFIX
      NSTEP = NSTEP + 1                                                 HYFIX
      IF (NSTEP.LT.100) GO TO 27                                        HYFIX
      WRITE(6,11) M,A,DEL,F1,F2,L,RM(1),RM(2),U,R,BD                    HYFIX
      STOP 104                                                          HYFIX
   27 IF (DABS(F2).LT.TEST) GO TO 50                                    HYFIX
```

```
      IF (F2) 35,50,30                                                HYFIX
   30 IF (DSIGN(ONE,A).EQ.DSIGN(ONE,A-DEL)) GO TO 32                  HYFIX
      A = A/2.0                                                       HYFIX
      DL = -A                                                         HYFIX
      GO TO 33                                                        HYFIX
   32 DL = -DEL                                                       HYFIX
      A = A - DEL                                                     HYFIX
   33 F1 = F2                                                         HYFIX
      GO TO 25                                                        HYFIX
   35 IF (F1.EQ.F2) GO TO 50                                          HYFIX
      A = A + F2*DL/(F1 - F2)                                         HYFIX
      IF (DABS(DEL).GT.TESD) GO TO 10                                 HYFIX
C                                                                     HYVAL
   50 RETURN                                                          HYVAL
      END                                                             HYVAL
```

```
      SUBROUTINE HYVBX(Q,S,B,M,RM)                                      HYVBX
C                                                      REV IV    02/07/87HYVBX
      IMPLICIT REAL*8(A-H,O-Z)                                          HYVBX
      DIMENSION Q(3),S(3),B(3),RM(2)                                    HYVBX
C FINDS LIMITS OF BOX IN DIRECTION Q,  Z = R*Q + S                      HYVBX
      LOGICAL VAL                                                       HYVBX
      M = 0                                                             HYVBX
      C = -1.0                                                          HYVBX
      DO 30 I = 1,3                                                     HYVBX
      IF(Q(I).EQ.0.0)GO TO 30                                           HYVBX
      DO 25 K = 1,2                                                     HYVBX
      VAL = .TRUE.                                                      HYVBX
      D = C*B(I) - S(I)                                                 HYVBX
      DO 10 J = 1,3                                                     HYVBX
      IF(J.EQ.I)GO TO 10                                                HYVBX
      IF(DABS(D*Q(J) + S(J)*Q(I)).GT.DABS(B(J)*Q(I)))VAL = .FALSE.      HYVBX
C     IF(DABS(R*Q(J) + S(J)).GT.B(J))VAL = .FALSE.                      HYVBX
   10 CONTINUE                                                          HYVBX
      IF(.NOT.VAL)GO TO 25                                              HYVBX
      R = D/Q(I)                                                        HYVBX
      IF(M.EQ.0)GO TO 20                                                HYVBX
      DO 15 L = 1,M                                                     HYVBX
      IF(R.EQ.RM(L)) GO TO 25                                           HYVBX
   15 CONTINUE                                                          HYVBX
   20 M = M + 1                                                         HYVBX
      RM(M) = R                                                         HYVBX
   25 C = -C                                                            HYVBX
   30 CONTINUE                                                          HYVBX
      IF(M.EQ.0)GO TO 35                                                HYVBX
      IF(RM(1).LT.RM(2))GO TO 35                                        HYVBX
      R = RM(1)                                                         HYVBX
      RM(1) = RM(2)                                                     HYVBX
      RM(2) = R                                                         HYVBX
   35 RETURN                                                            HYVBX
      END                                                               HYVBX
```

```
      DOUBLE PRECISION FUNCTION HYVFN(A,U,R,B,P)                      HYVFN
C                                                 REV IV   12/11/87HYVFN
      IMPLICIT REAL*8(A-H,O-Z)                                        HYVFN
      DIMENSION U(3),R(3),B(24)                                       HYVFN
      F = -1.0                                                        HYVFN
      DO 10 I = 1,3                                                   HYVFN
      Z = A*U(I) + R(I)                                               HYVFN
      C = B(I+16)                                                     HYVFN
      IF (P.GT.0.0) C = HYFCN(C,Z,B(I+1),P)                           HYVFN
   10 F = F + C*Z**2                                                  HYVFN
      HYVFN = F                                                       HYVFN
      RETURN                                                          HYVFN
      END                                                             HYVFN
```

```
      SUBROUTINE IMPLS2(MODE,J,H)                                       IMPLS2
C                                                        REV IV    07/24/86SLIP
C     CALLED BY SUBROUTINE UPDATE WHEN JOINT J LOCKS TO APPLY IMPULSE   IMPLS2
C     TO SET P.(D(M)'W(M) - D(N)'W(N)) = 0                              IMPLS2
C                                                                       IMPLS2
C     ARGUMENTS:                                                        IMPLS2
C          MODE -  0: FULL LOCK        P = I                            IMPLS2
C                  1: AXIS (H) FREE    P = I-HH'                        IMPLS2
C                 -1: AXIS (H) LOCKED  P = HH'                          IMPLS2
C                                                                       IMPLS2
C          J    - JOINT IDENTIFICATION NUMBER                           IMPLS2
C                                                                       IMPLS2
C          H    - AXIS VECTOR                                           IMPLS2
C                                                                       IMPLS2
C                                                                       IMPLS2
      IMPLICIT REAL*8(A-H,O-Z)                                          IMPLS2
      COMMON/CONTRL/ TIME,NSEG,NJNT,NPL,NBLT,NBAG,NVEH,NGRND,           IMPLS2
     *               NS,NQ,NSD,NFLX,NHRNSS,NWINDF,NJNTF,NPRT(36),NPG    PAGE
      COMMON/SGMNTS/ D(3,3,30),WMEG(3,30),WMEGD(3,30),U1(3,30),U2(3,30),IMPLS2
     *               SEGLP(3,30),SEGLV(3,30),SEGLA(3,30),NSYM(30)       IMPLS2
      COMMON/DESCRP/ PHI(3,30),W(30),RW(30),SR(4,60),HA(3,60),HB(3,60), SLIP
     *               RPHI(3,30),HT(3,3,60),SPRING(5,90),VISC(7,90),     IMPLS2
     *               JNT(30),IPIN(30),ISING(30),IGLOB(30),JOINTF(30)    IMPLS2
      COMMON/CMATRX/ V1(3,30),V2(3,30),V3(3,12),B12(3,3,60),A22(3,3,60),IMPLS2
     *               F(3,30),TQ(3,30),WJ(30),A11(3,3,30)                SLIP
      COMMON/CSTRNT/ A13(3,3,24),A23(3,3,24),B31(3,3,24),B32(3,3,24),   IMPLS2
     *               HHT(3,3,12),RK1(3,12),RK2(3,12),QQ(3,12),TQQ(3,12),IMPLS2
     *               RQQ(3,12),HQQ(3,12),SQQ(12),CFQQ(12),              IMPLS2
     *               KQ1(12),KQ2(12),KQTYPE(12)                         IMPLS2
      COMMON/FLXBLE/ HF(4,12,8),B42(3,3,24),V4(3,8),NFLEX(3,8)          IMPLS2
      COMMON/TEMPVS/ SM(3),SN(3),TM(3,3),TN(3,3),T(3,4),TT(3,4)         IMPLS2
      DIMENSION      TWA(3,3,30),TLA(3,3,30),H(3)                       IMPLS2
      CALL ELTIME(1,28)                                                 IMPLS2
      M = JNT(J)                                                        IMPLS2
      N = J+1                                                           IMPLS2
      DO 20 L=1,3                                                       IMPLS2
      DO 12 K=1,NGRND                                                   IMPLS2
      DO 12 I=1,3                                                       IMPLS2
      U1(I,K) = 0.0                                                     IMPLS2
   12 U2(I,K) = 0.0                                                     IMPLS2
      DO 13 K=1,NJNT                                                    IMPLS2
      DO 13 I=1,3                                                       IMPLS2
      V1(I,K) = 0.0                                                     IMPLS2
   13 V2(I,K) = 0.0                                                     IMPLS2
      IF (NQ.LE.0) GO TO 15                                             IMPLS2
      DO 14 K=1,NQ                                                      IMPLS2
      DO 14 I=1,3                                                       IMPLS2
   14 V3(I,K) = 0.0                                                     IMPLS2
   15 IF (NFLX.EQ.0) GO TO 18                                           IMPLS2
      DO 19 K=1,NFLX                                                    IMPLS2
```

```
      DO 19 I=1,3                                                       IMPLS2
   19 V4(I,K) = 0.0                                                     IMPLS2
   18 DO 16 I=1,3                                                       IMPLS2
      U2(I,M) = RPHI(I,M)*D(I,L,M)                                      IMPLS2
   16 U2(I,N) =-RPHI(I,N)*D(I,L,N)                                      IMPLS2
      CALL DAUX(L)                                                      IMPLS2
      DO 17 K=1,NGRND                                                   IMPLS2
      DO 17 I=1,3                                                       IMPLS2
      TLA(I,L,K) = SEGLA(I,K)                                           IMPLS2
   17 TWA(I,L,K) = WMEGD(I,K)                                           IMPLS2
   20 CONTINUE                                                          IMPLS2
      CALL DOT33(D(1,1,M),TWA(1,1,M),TM)                                IMPLS2
      CALL DOT33(D(1,1,N),TWA(1,1,N),TN)                                IMPLS2
      CALL DOT31(D(1,1,M),WMEG(1,M), SM)                                IMPLS2
      CALL DOT31(D(1,1,N),WMEG(1,N), SN)                                IMPLS2
      DO 22 I=1,3                                                       IMPLS2
      DO 21 K=1,3                                                       IMPLS2
      T(I,K) = TM(I,K) - TN(I,K)                                        IMPLS2
   21 TT(I,K) = T(I,K)                                                  IMPLS2
      T(I,4) = SN(I) - SM(I)                                            IMPLS2
   22 TT(I,4) = H(I)                                                    IMPLS2
      IF (MODE.GE.0) CALL DSMSOL(T,3,3)                                 IMPLS2
      IF (MODE.GT.0) CALL DSMSOL(TT,3,3)                                IMPLS2
      IF (MODE) 24,29,25                                                IMPLS2
   24 ST = 0.0                                                          IMPLS2
      STT = XDY(H,T,H)                                                  IMPLS2
      GO TO 26                                                          IMPLS2
   25 ST = 1.0                                                          IMPLS2
      STT = -(H(1)*TT(1,4) + H(2)*TT(2,4) + H(3)*TT(3,4))               IMPLS2
   26 STT =  (H(1)* T(1,4) + H(2)* T(2,4) + H(3)* T(3,4))/STT            IMPLS2
      DO 27 I=1,3                                                       IMPLS2
   27 T(I,4) = ST*T(I,4) + STT*TT(I,4)                                  IMPLS2
   29 DO 30 K=1,NGRND                                                   IMPLS2
      DO 30 I=1,3                                                       IMPLS2
      DO 30 L=1,3                                                       IMPLS2
      SEGLV(I,K) = SEGLV(I,K) + T(L,4)*TLA(I,L,K)                       IMPLS2
   30 WMEG (I,K) = WMEG (I,K) + T(L,4)*TWA(I,L,K)                       IMPLS2
      IF  (NPRT(3).NE.0)  CALL PRINT(6HIMPLS2)                          IMPLS2
      CALL ELTIME(2,28)                                                 IMPLS2
      RETURN                                                            IMPLS2
      END                                                               IMPLS2
```

```
      SUBROUTINE IMPULS(I1,I2,I3)                                       IMPULS
C                                                  REV IV    07/24/86SLIP
C         ARGUMENTS: I1 = 1 - IMPULS FOR PLELP.                         IMPULS
C                         3 - IMPULS FOR SEGSEG.                        IMPULS
C                         4 - IMPULS FOR VISPR OR EJOINT                IMPULS
C                    I2 = INDEX OF CONTACTING SEGMENT OR JOINT AXIS     IMPULS
C                    I3 = INDEX OF PLANE, SEGMENT OR JOINT AXIS         IMPULS
C                                                                       IMPULS
C                                                                       IMPULS
      IMPLICIT REAL*8 (A-H,O-Z)                                         IMPULS
      COMMON/CONTRL/ TIME,NSEG,NJNT,NPL,NBLT,NBAG,NVEH,NGRND,           IMPULS
     *               NS,NQ,NSD,NFLX,NHRNSS,NWINDF,NJNTF,NPRT(36),NPG    PAGE
      COMMON/SGMNTS/ D(3,3,30),WMEG(3,30),WMEGD(3,30),U1(3,30),U2(3,30),IMPULS
     *               SEGLP(3,30),SEGLV(3,30),SEGLA(3,30),NSYM(30)       IMPULS
      COMMON/DESCRP/ PHI(3,30),W(30),RW(30),SR(4,60),HA(3,60),HB(3,60), SLIP
     *               RPHI(3,30),HT(3,3,60),SPRING(5,90),VISC(7,90),     IMPULS
     *               JNT(30),IPIN(30),ISING(30),IGLOB(30),JOINTF(30)    IMPULS
      COMMON/CMATRX/ V1(3,30),V2(3,30),V3(3,12),B12(3,3,60),A22(3,3,60),IMPULS
     *               F(3,30),TQ(3,30),WJ(30),A11(3,3,30)                SLIP
      COMMON/JBARTZ/ MNPL(   30),MNBLT(    8),MNSEG(    30),MNBAG(     6), IMPULS
     *               MPL(3,5,30),MBLT(3,5,8),MSEG(3,5,30),MBAG(3,10,6), IMPULS
     *               NTPL( 5,30),NTBLT( 5,8),NTSEG( 5,30)               IMPULS
      COMMON/CSTRNT/ A13(3,3,24),A23(3,3,24),B31(3,3,24),B32(3,3,24),   IMPULS
     *               HHT(3,3,12),RK1(3,12),RK2(3,12),QQ(3,12),TQQ(3,12),IMPULS
     *               RQQ(3,12),HQQ(3,12),SQQ(12),CFQQ(12),              IMPULS
     *               KQ1(12),KQ2(12),KQTYPE(12)                         IMPULS
      COMMON/FLXBLE/ HF(4,12,8),B42(3,3,24),V4(3,8),NFLEX(3,8)          IMPULS
      COMMON/TEMPVI/ CREST,TTI(3),R1I(3),R2I(3),JSTOP(4,2,30)           IMPULS
      COMMON/TABLES/MXNTI,MXNTB,MXTB1,MXTB2,NTI(50),NTAB(1250),TAB(4500)DIMENB
      DIMENSION TEMP(3),DWR1(3),DWR2(3),DWR3(3),DWR4(3),VREL(3),DV(3)   IMPULS
      IF (TIME.EQ.0.0) GO TO 99                                         IMPULS
C                                                                       IMPULS
C     SPECIAL SETUP FOR CALL TO SUBROUTINE DAUX                         IMPULS
C     REPLACE SETUP WITH U1,U2,V1,V2,V3 = 0.                            IMPULS
C     ASSUME OTHER ARRAYS FROM PREVIOUS CALL TO DAUX.                   IMPULS
C                                                                       IMPULS
      CALL ELTIME(1,27)                                                 IMPULS
      CALL OUTPUT(0)                                                    IMPULS
      KQTEST = 0                                                        IMPULS
      NT = 0                                                            IMPULS
      IF (I1.EQ.1) NT = NTPL (I2,I3)                                    IMPULS
      IF (I1.EQ.3) NT = NTSEG(I2,I3)                                    IMPULS
      IF (NT.EQ.0) GO TO 29                                             IMPULS
      KQ = -NTAB(NT+1)                                                  IMPULS
      IF (KQ.LE.0) GO TO 29                                             IMPULS
      KQTYPE(KQ) = IABS(KQTYPE(KQ))                                     IMPULS
      CALL DAUX(0)                                                      IMPULS
   29 IF (NQ.LE.0) GO TO 31                                             IMPULS
      DO 30 J=1,NQ                                                      IMPULS
      DO 30 I=1,3                                                       IMPULS
```

```
   30 V3(I,J) = 0.0                                                 IMPULS
   31 DO 32 J=1,NGRND                                               IMPULS
      DO 32 I=1,3                                                   IMPULS
      U1(I,J) = 0.0                                                 IMPULS
   32 U2(I,J) = 0.0                                                 IMPULS
      IF (NJNT.LE.0) GO TO 21                                       IMPULS
      DO 33 J=1,NJNT                                                IMPULS
      DO 33 I=1,3                                                   IMPULS
      V1(I,J) = 0.0                                                 IMPULS
   33 V2(I,J) = 0.0                                                 IMPULS
   21 IF (NFLX.EQ.0) GO TO 23                                       IMPULS
      DO 22 J=1,NFLX                                                IMPULS
      DO 22 I=1,3                                                   IMPULS
   22 V4(I,J) = 0.0                                                 IMPULS
C                                                                   IMPULS
C     REPLACE CALLS TO CONTACT AND VISPR WITH SINGLE CALL           IMPULS
C     AT FIRST CONTACT IF NOT CONSTRAINT.                           IMPULS
C                                                                   IMPULS
   23 IF (I1.NE.1) GO TO 34                                         IMPULS
      NT = NTPL(I2,I3)                                              IMPULS
      M1 = MPL(1,I2,I3)                                             IMPULS
      M2 = MPL(2,I2,I3)                                             IMPULS
      M3 = MPL(3,I2,I3)                                             IMPULS
      CALL PLELP(M2,M3,M1,I3,NT)                                    IMPULS
      IF (NTAB(NT+1).LT.0) GO TO 37                                 IMPULS
      K1 = M2                                                       IMPULS
      K2 = M1                                                       IMPULS
      GO TO 39                                                      IMPULS
   34 IF (I1.NE.3) GO TO 35                                         IMPULS
      NT = NTSEG(I2,I3)                                             IMPULS
      M1 = MSEG(1,I2,I3)                                            IMPULS
      M2 = MSEG(2,I2,I3)                                            IMPULS
      M3 = MSEG(3,I2,I3)                                            IMPULS
      CALL SEGSEG(I3,M1,M2,M3,NT)                                   IMPULS
      IF (NTAB(NT+1).LT.0) GO TO 37                                 IMPULS
      K1 = I3                                                       IMPULS
      K2 = M2                                                       IMPULS
      GO TO 39                                                      IMPULS
   35 IF (I1.NE.4) WRITE (6,36) I1,I2,I3                            IMPULS
   36 FORMAT('0 IMPROPER ARGUMENTS TO SUBROUTINE IMPULS'/           IMPULS
     *         '  ARGUMENTS = ', 3I6 /                              IMPULS
     *         '  PROGRAM TERMINATED' )                             IMPULS
      IF  (I1.NE.4)  STOP 33                                        IMPULS
C                                                                   IMPULS
C     RECALL VISPR FOR JOINT STOP.                                  IMPULS
C                                                                   IMPULS
      IF (IABS(IPIN(I3)).NE.4) GO TO 25                             IMPULS
      CALL EJOINT(I2,I3)                                            IMPULS
      GO TO 26                                                      IMPULS
   25 CALL VISPR(I2,I3)                                             IMPULS
```

```
   26 K1 = IABS(JNT(I3))                                                    IMPULS
      K2 = I3+1                                                             IMPULS
      GO TO 39                                                              IMPULS
C                                                                           IMPULS
C     SET UP SPECIAL U1,U2 FOR FIRST CONTACT OF CONSTRAINT.                 IMPULS
C                                                                           IMPULS
   37 KQ = -NTAB(NT+1)                                                      IMPULS
      KQTEST = 1                                                            IMPULS
      KQTYPE(KQ) = -IABS(KQTYPE(KQ))                                        IMPULS
      K1 = KQ1(KQ)                                                          IMPULS
      K2 = KQ2(KQ)                                                          IMPULS
      IF (K1.GT.NSEG) GO TO 38                                              IMPULS
      CALL MAT31(A13(1,1,2*KQ-1),QQ(1,KQ),U1(1,K1))                         IMPULS
      CALL MAT31(A23(1,1,2*KQ-1),QQ(1,KQ),U2(1,K1))                         IMPULS
   38 IF (K2.GT.NSEG) GO TO 39                                              IMPULS
      CALL MAT31(A13(1,1,2*KQ  ),QQ(1,KQ),U1(1,K2))                         IMPULS
      CALL MAT31(A23(1,1,2*KQ  ),QQ(1,KQ),U2(1,K2))                         IMPULS
C                                                                           IMPULS
C     FINAL SETUP OF U1 AND U2                                              IMPULS
C                                                                           IMPULS
   39 DO 40 J=1,NGRND                                                       IMPULS
      DO 40 I=1,3                                                           IMPULS
      U1(I,J) = U1(I,J)*RW(J)                                               IMPULS
   40 U2(I,J) = U2(I,J)*RPHI(I,J)                                           IMPULS
      CALL DAUX(I1)                                                         IMPULS
      IF (KQTEST.EQ.1) KQTYPE(KQ) = IABS(KQTYPE(KQ))                        IMPULS
      IF (NPRT(10).NE.0) CALL PRINT(6HPREIMP)                               IMPULS
      IF (I1.GT.3) GO TO 51                                                 IMPULS
      IF (NPRT(10).NE.0) WRITE (6,42) R1I,R2I                               IMPULS
   42 FORMAT ('0'/(6G20.8))                                                 IMPULS
      CALL CROSS(WMEG (1,K1),R1I(1),TEMP)                                   IMPULS
      CALL DOT31(D(1,1,K1),TEMP,DWR1(1))                                    IMPULS
      CALL CROSS(WMEG (1,K2),R2I(1),TEMP)                                   IMPULS
      CALL DOT31(D(1,1,K2),TEMP,DWR2(1))                                    IMPULS
      CALL CROSS(WMEGD(1,K1),R1I(1),TEMP)                                   IMPULS
      CALL DOT31(D(1,1,K1),TEMP,DWR3(1))                                    IMPULS
      CALL CROSS(WMEGD(1,K2),R2I(1),TEMP)                                   IMPULS
      CALL DOT31(D(1,1,K2),TEMP,DWR4(1))                                    IMPULS
      TVREL = 0.0                                                           IMPULS
      TDV   = 0.0                                                           IMPULS
      DO 50 I=1,3                                                           IMPULS
      VREL(I) = SEGLV(I,K1)+DWR1(I) - SEGLV(I,K2)-DWR2(I)                   IMPULS
      DV  (I) = SEGLA(I,K1)+DWR3(I) - SEGLA(I,K2)-DWR4(I)                   IMPULS
      TVREL = TVREL + TTI(I)*VREL(I)                                        IMPULS
   50 TDV   = TDV   + TTI(I)*DV  (I)                                        IMPULS
      GO TO 53                                                              IMPULS
   51 CALL DOT31(D(1,1,K1),WMEG (1,K1),DWR1(1))                             IMPULS
      CALL DOT31(D(1,1,K2),WMEG (1,K2),DWR2(1))                             IMPULS
      CALL DOT31(D(1,1,K1),WMEGD(1,K1),DWR3(1))                             IMPULS
      CALL DOT31(D(1,1,K2),WMEGD(1,K2),DWR4(1))                             IMPULS
```

```
      TVREL = 0.0                                                        IMPULS
      TDV   = 0.0                                                        IMPULS
      DO 52 I=1,3                                                        IMPULS
      VREL(I) = DWR1(I) - DWR2(I)                                        IMPULS
      DV  (I) = DWR3(I) - DWR4(I)                                        IMPULS
      TVREL = TVREL + TTI(I)*VREL(I)                                     IMPULS
   52 TDV   = TDV   + TTI(I)*DV  (I)                                     IMPULS
   53 ALPHA = 0.0                                                        IMPULS
C                                                                        IMPULS
C     NOTE: CREST IS SUPPLIED AS (1+E)/2 WHERE E IS THE CLASSICAL        IMPULS
C     COEFFICIENT OF RESTITUTION BUT WITH A RANGE OF -1 TO +1.           IMPULS
C     CREST HAS A RANGE OF 0 TO +1 WHERE 0 (E=-1) REPRESENTS NO IMPULSE.IMPULS
C                                                                        IMPULS
      IF (TDV.NE.0.0) ALPHA = -2.0*CREST*TVREL/TDV                       IMPULS
      IF (NPRT(10).NE.0) WRITE (6,42) DWR1,DWR2,DWR3,DWR4,               IMPULS
     *                                TTI,VREL,DV,                       IMPULS
     *                                TVREL,TDV,CREST,ALPHA              IMPULS
      DO 60 J=1,NGRND                                                    IMPULS
      DO 60 I=1,3                                                        IMPULS
      SEGLV(I,J) = SEGLV(I,J) + ALPHA*SEGLA(I,J)                         IMPULS
   60 WMEG (I,J) = WMEG (I,J) + ALPHA*WMEGD(I,J)                         IMPULS
      IF  (NPRT(10).NE.0)  CALL OUTPUT(1)                                IMPULS
      IF  (NPRT( 3).NE.0)  CALL PRINT(6HIMPULS)                          IMPULS
      CALL ELTIME(2,27)                                                  IMPULS
   99 RETURN                                                             IMPULS
      END                                                                IMPULS
```

```
      SUBROUTINE INITAL                                                 INITAL
C                                                  REV IV    07/24/86SLIP
C     PERFORMS CARD INPUT AND COMPUTATIONS FOR INITIAL                  INITAL
C     POSITIONING OF THE CRASH VICTIM'S BODY SEGMENTS.                  INITAL
C                                                                       INITAL
      IMPLICIT REAL*8(A-H,O-Z)                                          INITAL
      COMMON/CONTRL/ TIME,NSEG,NJNT,NPL,NBLT,NBAG,NVEH,NGRND,           INITAL
     *               NS,NQ,NSD,NFLX,NHRNSS,NWINDF,NJNTF,NPRT(36),NPG    PAGE
      COMMON/SGMNTS/ D(3,3,30),WMEG(3,30),WMEGD(3,30),U1(3,30),U2(3,30),INITAL
     *               SEGLP(3,30),SEGLV(3,30),SEGLA(3,30),NSYM(30)       INITAL
      COMMON/DESCRP/ PHI(3,30),W(30),RW(30),SR(4,60),HA(3,60),HB(3,60), SLIP
     *               RPHI(3,30),HT(3,3,60),SPRING(5,90),VISC(7,90),     INITAL
     *               JNT(30),IPIN(30),ISING(30),IGLOB(30),JOINTF(30)    INITAL
      COMMON/VPOSTN/ ZPLT(3),SPLT(3),AXV(3,6),VATAB(6,501,6),           VEHICL
     *               VTO(6),VDT(6),TIMEV(6),OMEGV(6),NVTAB(6),INDXV(6)  INITAL
      COMMON/TITLES/ DATE(3),COMENT(40),VPSTTL(20),BDYTTL(5),           INITAL
     *               BLTTTL(5,8),PLTTL(5,30),BAGTTL(5,6),SEG(30),       INITAL
     *               JOINT(30),CGS(30),JS(30)                           INITAL
      COMMON/CEULER/ IEULER(30),HIR(3,3,90),ANG(3,30),ANGD(3,30),       JDRIFT
     *               FE(3,30),TQE(3,30),CONST(5,30)                     JDRIFT
      REAL DATE,COMENT,VPSTTL,BDYTTL,BLTTTL,PLTTL,BAGTTL,SEG,JOINT      INITAL
      LOGICAL*1 CGS,JS                                                  INITAL
      COMMON/CNSNTS/ PI,RADIAN,G,THIRD,EPS(24),                         INITAL
     *               UNITL,UNITM,UNITT,GRAVTY(3),TWOPI                  TWOPI
      COMMON/TEMPVS/ TMP(140),WMGDEG(3,30),T(3),S(3),A(3,2),Z(3,3)      SLIP
C                         NOTE : CHAIN ALSO USES TEMPVS.                INITAL
      DIMENSION  YPR(3,30) , IYPR(4,30)                                 INITAL
C                                                                       INITAL
C     INPUT CARD G.1.A (PLOT COORDINATES OF VEHICLE REFERENCE ORIGIN)   INITAL
C                                                                       INITAL
      READ(5,22) ZPLT,I1,J1,I2,J2,I3                                    INITAL
   22 FORMAT(3F10.0,5I4)                                                INITAL
      S(1) = 10.0                                                       INITAL
      S(2) =  6.0                                                       INITAL
      S(3) =  1.0                                                       INITAL
C                                                                       INITAL
C     IF J1*0, INPUT CARD G.1.B (PLOT SCALING INPUT)                    INITAL
C                                                                       INITAL
      IF  (J1.NE.0)  READ (5,22) S                                      INITAL
      SPLT(1) = 1.0/S(3)                                                INITAL
      SPLT(2) = 1.0/S(3)                                                INITAL
      SPLT(3) = -(S(1)/S(2))/S(3)                                       INITAL
      WRITE  (6,23)  NPG,ZPLT,I1,J1,I2,J2,I3,S                          PAGE
      NPG=NPG+1                                                         PAGE
   23 FORMAT('1 SUBROUTINE INITAL INPUT',98X,'PAGE',I5/120X,'CARD G.1'/ PAGE
     *   '     ZPLT(X)   ZPLT(Y)   ZPLT(Z)    I1     J1     I2     J2     I3',INITAL
     *   '     SPLT(1)   SPLT(2)   SPLT(3)'/3F10.0,5I6,3F10.2)          INITAL
C                                                                       INITAL
C     INPUT CARDS G.2.A - G.2.N                                         INITAL
C                                                                       INITAL
```

```
C     INITIAL LINEAR POSITION (IN) AND (IF I3=1) VELOCITY (IN/SEC)      INITAL
C     OF EACH BASE BODY SEGMENT. IF I3=0, VELOCITY WILL BE SET TO       INITAL
C     INITIAL VELOCITY OF VEHICLE. INPUTS IN INERTIAL REFERENCE.        INITAL
C                                                                       INITAL
      DO 37 J=1,NSEG                                                    INITAL
      IF(J.GT.1.AND.IABS(JNT(J-1)).GT.0) GO TO 37                       INITAL
      READ(5,24) (SEGLP(I,J),I=1,3),(SEGLV(I,J),I=1,3)                  INITAL
   24 FORMAT (6F10.0 , 4I3)                                             INITAL
      IF(I3.GT.0) GO TO 37                                              INITAL
      DO 36 I=1,3                                                       INITAL
   36 SEGLV(I,J) = SEGLV(I,NVEH)                                        INITAL
   37 CONTINUE                                                          INITAL
C                                                                       INITAL
C     INPUT CARDS G.3.A - G.3.N                                         INITAL
C                                                                       INITAL
C     FOR EACH BODY SEGMENT SUPPLY YAW, PITCH AND ROLL (DEGREES)        INITAL
C     AND (IF I3=1) THE ANGULAR VELOCITY IN LOCAL REFERENCE (DEG/SEC).  INITAL
C     IF I3=0, THE ANGULAR VELOCITY (BLANK ON INPUT CARDS) WILL BE SET  INITAL
C     EQUAL TO THE INITIAL ANGULAR VELOCITY OF THE VEHICLE.             INITAL
C                                                                       INITAL
      FIRST = 0.0                                                       INITAL
      DO 40 J=1,NSEG                                                    INITAL
      READ (5,24) (YPR(I,J),I=1,3),(WMGDEG(I,J),I=1,3),(IYPR(I,J),I=1,4)INITAL
      ID1 = IYPR(1,J)                                                   INITAL
      DO 38 I=1,3                                                       INITAL
      IF (ID1.EQ.0) IYPR(I,J) = I                                       INITAL
   38 WMEG(I,J) = WMGDEG(I,J)*RADIAN                                    INITAL
      IF  (ID1.GE.0)  GO TO 60                                          INITAL
C                                                                       INITAL
C     READ CARD G.3.J2 FOR SEGMENT NO. J WHEN IYPR(1,J) IS NEGATIVE.    INITAL
C                                                                       INITAL
      READ  (5,24)  A,II,IK,JJ,JK                                       INITAL
      IJ = II                                                           INITAL
      LK = IK                                                           INITAL
      DO  54  K=1,2                                                     INITAL
      IF  (IJ.GT.0)  GO TO 52                                           INITAL
      DO  51  I=1,3                                                     INITAL
   51 Z(I,LK) = A(I,K)                                                  INITAL
      GO TO 53                                                          INITAL
   52 DA1 = A(1,K)*RADIAN                                               INITAL
      DA2 = A(2,K)*RADIAN                                               INITAL
      SA1 = DSIN(DA1)                                                   INITAL
      SA2 = DSIN(DA2)                                                   INITAL
      CA1 = DCOS(DA1)                                                   INITAL
      CA2 = DCOS(DA2)                                                   INITAL
      IJ1 = IJ+1                                                        INITAL
      IJ2 = IJ+2                                                        INITAL
      IF  (IJ1.GT.3)  IJ1= IJ1-3                                        INITAL
      IF  (IJ2.GT.3)  IJ2= IJ2-3                                        INITAL
      SGN = 1.0                                                         INITAL
```

```
      IF  (SA1.LT.0.0 .AND. CA2.LT.0.0)  SGN = -1.0                     INITAL
      Z(IJ ,LK) = SGN*SA1*CA2                                           INITAL
      Z(IJ1,LK) = SGN*SA1*SA2                                           INITAL
      Z(IJ2,LK) = SGN*CA1*CA2                                           INITAL
   53 IJ = JJ                                                           INITAL
   54 LK = JK                                                           INITAL
      ZDOTIJ = Z(1,IK)*Z(1,JK) + Z(2,IK)*Z(2,JK) + Z(3,IK)*Z(3,JK)      INITAL
      ZDOTII = Z(1,IK)*Z(1,IK) + Z(2,IK)*Z(2,IK) + Z(3,IK)*Z(3,IK)      INITAL
      RATIO = ZDOTIJ/ZDOTII                                             INITAL
      DO  55  I=1,3                                                     INITAL
   55 Z(I,JK) = Z(I,JK) - RATIO*Z(I,IK)                                 INITAL
      LK = 6-IK-JK                                                      INITAL
      IT = MOD(JK-IK+3,3)                                               INITAL
      IF  (IT.EQ.1)  CALL CROSS(Z(1,IK),Z(1,JK),Z(1,LK))                INITAL
      IF  (IT.EQ.2)  CALL CROSS(Z(1,JK),Z(1,IK),Z(1,LK))                INITAL
      DO  57  K=1,3                                                     INITAL
      IYPR(K,J) = 4-K                                                   INITAL
      SUM = 0.0                                                         INITAL
      DO  56  I=1,3                                                     INITAL
   56 SUM = SUM + Z(I,K)**2                                             INITAL
      SQUM = DSQRT(SUM)                                                 INITAL
      DO  57  I=1,3                                                     INITAL
   57 D(K,I,J) = Z(I,K)/SQUM                                            INITAL
      CALL YPRDEG (D(1,1,J),YPR(1,J))                                   INITAL
      IF  (FIRST.EQ.0.0)  WRITE (6,58)                                  INITAL
   58 FORMAT('0 INITIAL ANGULAR ROTATIONS COMPUTED FROM CARDS G.3.J2'// INITAL
     *       '  SEGMENT',10X,'SEGMENT PRIMARY AXIS',                    INITAL
     *       12X,'SEGMENT SECONDARY AXIS',30X,'ANGULAR ROTATIONS (DEG)'/INITAL
     *       '  NO. SEG',9X,'A1',8X,'A2',8X,'A3',11X,'B1',8X,'B2',8X,   INITAL
     *       'B3',7X,'II  IK  JJ  JK',9X,'YAW',6X,'PITCH',5X,'ROLL'/)   INITAL
      FIRST = 1.0                                                       INITAL
      WRITE (6,59) J,SEG(J),A,II,IK,JJ,JK,(YPR(I,J),I=1,3)              INITAL
   59 FORMAT (I4,1X,A4,3X,3F10.3,3X,3F10.3,3X,4I4,3X,3F10.3)            INITAL
   60 M = IYPR(4,J)                                                     INITAL
      IF (M.EQ.0) M=NGRND                                               INITAL
      IF  (M.GE.J .AND. M.LE.NSEG)  STOP 24                             INITAL
      IF (J.EQ.1) GO TO 80                                              VAXCHG
      IF  (M.LT.0 .AND. -M.NE.IABS(JNT(J-1)))  STOP 25                  INITAL
   80 CALL DRCIJK (D,YPR,IYPR,HT,J)                                     VAXCHG
      IF (I3.GT.0) GO TO 40                                             INITAL
      CALL DOT31(D(1,1,NVEH),WMEG(1,NVEH),T)                            INITAL
      CALL MAT31(D(1,1,J),T,WMEG(1,J))                                  INITAL
      DO 39 I=1,3                                                       INITAL
   39 WMGDEG(I,J) = WMEG(I,J)/RADIAN                                    INITAL
   40 CONTINUE                                                          INITAL
      CALL VEHPOS                                                       INITAL
      IF(NJNT.EQ.0) GOTO 41                                             JDRIFT
      CALL CHAIN(0)                                                     JDRIFT
      CALL EJOINT(1,0)                                                  JDRIFT
      DO 62 J=1,NJNT                                                    JDRIFT
```

```
      IF(IABS(IPIN(J)).NE.4) GOTO 62                                    JDRIFT
      IF(IEULER(J).NE.2) GOTO 62                                        JDRIFT
      DA1 = ANG(2,J) + CONST(2,J)                                       JDRIFT
      CONST(4,J) = DCOS(DA1)                                            JDRIFT
      CONST(5,J) = DSIN(DA1)                                            JDRIFT
   62 CONTINUE                                                          JDRIFT
C                                                                       INITAL
C     OUTPUT INITIAL BODY SEGMENT POSITIONS.                            INITAL
C                                                                       INITAL
   41 WRITE (6,42) UNITL,UNITL,UNITT                                    JDRIFT
   42 FORMAT('0 INITIAL POSITIONS (INERTIAL REFERENCE)',70X,'CARDS G.2'/INITAL
     *      /'  SEGMENT',11X,'LINEAR POSITION (',A4,')',                INITAL
     *       14X,'LINEAR VELOCITY (',A4,'/',A4,')'/                     AFREVS
     *       '  NO. SEG',2(9X,'X',11X,'Y',11X,'Z',5X) )                 INITAL
      WRITE (6,43) (J,SEG(J),(SEGLP(I,J),I=1,3),(SEGLV(I,J),I=1,3)      INITAL
     *             ,J=1,NSEG)                                           INITAL
   43 FORMAT(I4,1X,A4,3X,3F12.5,3X,3F12.5)                              INITAL
      WRITE (6,44) UNITT                                                INITAL
   44 FORMAT('0 INITIAL ANGULAR ROTATION AND VELOCITY',71X,'CARDS G.3'//INITAL
     *       '  SEGMENT',11X,'ANGULAR ROTATION (DEG)',                  AFREVS
     *       14X,'ANGULAR VELOCITY (DEG/',A4,')'/                       INITAL
     *       '  NO. SEG',8X,'YAW',8X,'PITCH',7X,'ROLL',                 INITAL
     *       13X,'X',11X,'Y',11X,'Z',15X,'IYPR' )                       INITAL
      WRITE (6,46) (J,SEG(J),(YPR(I,J),I=1,3),(WMGDEG(I,J),I=1,3),      INITAL
     *             (IYPR(I,J),I=1,4),J=1,NSEG)                          INITAL
   46 FORMAT(I4,1X,A4,3X,3F12.5,3X,3F12.5,3X,4I4)                       INITAL
      IF (I3.EQ.0) WRITE (6,45)                                         INITAL
   45 FORMAT('0 LINEAR AND ANGULAR VELOCITIES HAVE BEEN SET EQUAL TO THEINITAL
     * INITIAL VEHICLE VELOCITIES.')                                    INITAL
      IF  (NHRNSS.NE.0)  CALL HBPLAY                                    INITAL
      IF  (I1.EQ.15)  CALL EQUILB (YPR,IYPR)                            INITAL
      CALL UNIT1(0)                                                     JDRIFT
      CALL ROTATE                                                       INITAL
      CALL ELTIME(2,2)                                                  INITAL
      RETURN                                                            INITAL
      END                                                               INITAL
```

```
      SUBROUTINE INTERS(A,B,XM,T,X,V,AX)                                INTERS
C                                                   REV IV    07/23/86TWOPI
C     DETERMINES INTERSECTION OF ELLIPSOIDS                             INTERS
C         X'AX = 1                                                      INTERS
C         (X'-M')B(X-M) = 1                                             INTERS
C     WHERE A AND B ARE ELLIPSOID MATRICES                              INTERS
C     IF T ENTERS AS +1.0 , A IS EXTERNAL TO B AND                      INTERS
C                 AS -1.0 , A IS INTERNAL TO B.                         INTERS
C                                                                       INTERS
C     IF V ENTERS AS NON-ZERO, WILL USE PREVIOUS VALUE FOR START.       INTERS
C     (AX) RETURNS AS (A)*(X).                                          INTERS
C                                                                       INTERS
C     RETURNS T>1 - NO INTERSECTION                                     INTERS
C             T<1 - INTERSECTION IN WHICH CASE X WILL                   INTERS
C                   CONTAIN COORDINATES OF CONTACT OF                   INTERS
C                   CONTRACTED ELLIPSOIDS.                              INTERS
C                                                                       INTERS
      IMPLICIT REAL*8 (A-H,O-Z)                                         INTERS
      DIMENSION A(3,3),B(3,3),XM(3),X(3)                                INTERS
      DIMENSION C(3,4),Z(3),BM(3),AX(3),AM(3)                           INTERS
      EQUIVALENCE (Z(1),C(1,4))                                         INTERS
      COMMON/CNSNTS/ PI,RADIAN,G,THIRD,EPS(24),                         INTERS
     *               UNITL,UNITM,UNITT,GRAVTY(3),TWOPI                  TWOPI
C                               INITIALIZATION                          INTERS
C                                   EVALUATE BM,M'AM,M'BM               INTERS
C                                   SET N=0, V=M'BM/M'AM                INTERS
      N = 0                                                             INTERS
      BMM = 0.0                                                         INTERS
      AMM = 0.0                                                         INTERS
      DO 11 I=1,3                                                       INTERS
      BM(I) = 0.0                                                       INTERS
      AM(I) = 0.0                                                       INTERS
      DO 10 J=1,3                                                       INTERS
      IF (DABS(A(I,J)).LT.EPS(20)) A(I,J) = 0.0                         INTERS
      AM(I) = AM(I) + A(I,J)*XM(J)                                      INTERS
      IF (DABS(B(I,J)).LT.EPS(20)) B(I,J) =0.0                          INTERS
   10 BM(I) = BM(I) + B(I,J)*XM(J)                                      INTERS
      BMM = BMM + XM(I)*BM(I)                                           INTERS
   11 AMM = AMM + XM(I)*AM(I)                                           INTERS
      IF (V.EQ.0.0) V=T*DSQRT(BMM/AMM)                                  INTERS
      IDONE = 0                                                         INTERS
C                               NEWTON-RAPHSON ITERATION FOR            INTERS
C                               G(V) = FA(V)-FB(V) = 0                  INTERS
C                                   SOLVE (VA+B)X = BM  FOR X           INTERS
      ITER = 0                                                          INTERS
   20 ITER = ITER+1                                                     INTERS
      DO 22 I=1,3                                                       INTERS
      DO 21 J=1,3                                                       INTERS
   21 C(I,J) = V*A(I,J) + B(I,J)                                        INTERS
   22 Z(I)   = BM(I)                                                    INTERS
```

```
      CALL DSMSOL(C,3,3)                                              INTERS
C                                       EVALUATE AX                   INTERS
C                                                FA(V) = X'AX         INTERS
C                                                FB(V) = -V(X'-M')AX  INTERS
      FA = 0.0                                                        INTERS
      FB = 0.0                                                        INTERS
      CALL MAT31(A,Z,AX)                                              INTERS
      DO 30 I=1,3                                                     INTERS
      X(I) = Z(I)                                                     INTERS
      FA = FA+X(I)*AX(I)                                              INTERS
   30 FB = FB+(X(I)-XM(I))*AX(I)                                      INTERS
      FB = -V*FB                                                      INTERS
      IF (T.LT.0.0) FA = 1.0/FA                                       INTERS
      IF (IDONE.EQ.1) GO TO 60                                        INTERS
C                                       TEST FOR INTERSECTION         INTERS
      IF (FA-FB) 32,60,31                                             INTERS
C                                       IF FA>FB>1, NO INTERSECTION   INTERS
   31 IF (T.GT.0.0.AND.FB.LT.1.0) GO TO 40                            INTERS
      IF (T.LT.0.0.AND.FA.GT.1.0) GO TO 40                            INTERS
      IF (N.EQ.0) GO TO 60                                            INTERS
      GO TO 62                                                        INTERS
C                                       IF FA<FB<1 , INTERSECTION     INTERS
   32 IF (T.GT.0.0.AND.FB.LE.1.0) N=1                                 INTERS
      IF (T.LT.0.0.AND.FA.GE.1.0) N=1                                 INTERS
C                                       SOLVE (VA+B)Z = AX  FOR Z     INTERS
   40 DO 42 I=1,3                                                     INTERS
      DO 41 J=1,3                                                     INTERS
   41 C(I,J) = V*A(I,J) + B(I,J)                                      INTERS
   42 Z(I)   = AX(I)                                                  INTERS
      CALL DSMSOL(C,3,3)                                              INTERS
C                                       F'A(V) = -2X'AZ               INTERS
      CALL MAT31 (A,Z,AX)                                             INTERS
      FPA = X(1)*AX(1)                                                INTERS
     *    + X(2)*AX(2)                                                INTERS
     *    + X(3)*AX(3)                                                INTERS
      FPA = -(FPA+FPA)                                                INTERS
C                                       DV = -G(V)/G'(V)              INTERS
      DV = 1.0 + V                                                    INTERS
      IF (T.LT.0.0) DV = V-FA**2                                      INTERS
      DV = (FB-FA)/(DV*FPA)                                           INTERS
      IF (ITER.GE.50) GO TO 62                                        INTERS
C                                       TEST FOR CONVERGENCE          INTERS
      IF (T*(V+DV).LE.0.0) DV = -0.5*V                                INTERS
      V    = V+DV                                                     INTERS
      DV = DABS(DV/V)                                                 INTERS
      IF (DV.LE.EPS(12)) IDONE=1                                      INTERS
      GO TO 20                                                        INTERS
C                                  FA(V) = FB(B), RETURN              INTERS
   60 IF (T.LT.0.0) FA = 1.0/FB                                       INTERS
      T = DSQRT(FA)                                                   INTERS
```

```
      IF (FA.GT.1.0) GO TO 61                                          INTERS
      N = 1                                                            INTERS
      GO TO 71                                                         INTERS
   61 IF (N.EQ.0) GO TO 71                                             INTERS
   62 WRITE (6,63)                                                     INTERS
   63 FORMAT(' INTERS ITERATION DID NOT CONVERGE')                     INTERS
   71 CONTINUE                                                         INTERS
      RETURN                                                           INTERS
      END                                                              INTERS
```

```
      SUBROUTINE KINPUT                                                 KINPUT
C                                                REV IV    07/23/86TWOPI
C     PERFORMS THE FOLLOWING CARD INPUT AFTER CARDS E.1-E.4 (SUBROUTINE KINPUT
C     CINPUT) AND BEFORE CARDS F.1-F.5 (SUBROUTINE FINPUT).             KINPUT
C         CARD E.5 -  NO LONGER REQUIRED                                WINDOP
C         CARDS E.6 - DEFINITIONS OF WIND FORCE FUNCTIONS AND DRAG      WINDOP
C                     COEFFICIENT FUNCTIONS                             WINDOP
C         CARDS E.7 - DEFINITIONS OF JOINT RESTORING FORCE FUNCTIONS    KINPUT
C                                                                       KINPUT
      IMPLICIT REAL*8(A-H,O-Z)                                          KINPUT
      COMMON/CONTRL/ TIME,NSEG,NJNT,NPL,NBLT,NBAG,NVEH,NGRND,           KINPUT
     *               NS,NQ,NSD,NFLX,NHRNSS,NWINDF,NJNTF,NPRT(36),NPG    PAGE
      COMMON/TABLES/MXNTI,MXNTB,MXTB1,MXTB2,NTI(50),NTAB(1250),TAB(4500)DIMENB
      COMMON/CNSNTS/ PI,RADIAN,G,THIRD,EPS(24),                         KINPUT
     *               UNITL,UNITM,UNITT,GRAVTY(3),TWOPI                  TWOPI
      COMMON/TITLES/ DATE(3),COMENT(40),VPSTTL(20),BDYTTL(5),           WINDOP
     *               BLTTTL(5,8),PLTTL(5,30),BAGTTL(5,6),SEG(30),       WINDOP
     *               JOINT(30),CGS(30),JS(30)                           WINDOP
      REAL DATE,COMENT,VPSTTL,BDYTTL,BLTTTL,PLTTL,BAGTTL,SEG,JOINT      TGMOD1
      COMMON/TEMPVS/ JTITLE(5,51),NF(5),MS(3),KTITLE(31),TH(50)         KINPUT
C     NOTE: TEMPVS IS SHARED HERE WITH SUBROUTINES CINPUT AND FINPUT.   KINPUT
      REAL BLANK,JTITLE,KTITLE                                          KINPUT
      DATA BLANK/4H    /                                                KINPUT
   11 FORMAT(2I6)                                                       KINPUT
      J1 = MXTB1+1                                                      KINPUT
      IF (NWINDF.LE.0) GO TO 31                                         KINPUT
      DO 30 K=1,NWINDF                                                  KINPUT
C                                                                       KINPUT
C     INPUT CARD E.6.A - FUNCTION NO. AND TITLE                         KINPUT
C                                                                       KINPUT
      READ (5,12) I,(KTITLE(J),J=1,5)                                   KINPUT
   12 FORMAT(I4,4X,5A4)                                                 KINPUT
      WRITE (6,13) I,(KTITLE(J),J=1,5),I,J1,NPG                         PAGE
      NPG=NPG+1                                                         PAGE
   13 FORMAT('1 WIND FORCE FUNCTION NO.',I4,4X,5A4,10X,'NTI(',I2,') =', KINPUT
     *        I5,46X,'PAGE',I5/120X,'CARDS E.6'/)                       PAGE
      IF (I.LE.0.OR.I.GT.50) WRITE (6,14)                               KINPUT
   14 FORMAT('0 IMPROPER FUNCTION NO. PROGRAM TERMINATED.')             KINPUT
      IF (I.LE.0.OR.I.GT.50) STOP 11                                    KINPUT
      IF (NTI(I).NE.0) WRITE (6,15) I                                   KINPUT
   15 FORMAT('0 FUNCTION NO.',I4,' HAS ALREADY BEEN INPUTTED AND WILL BEKINPUT
     * REPLACED BY THIS FUNCTION.')                                     KINPUT
      NTI(I) = J1                                                       KINPUT
      DO 16 J=1,5                                                       KINPUT
   16 JTITLE(J,I) = KTITLE(J)                                           KINPUT
      J2 = J1+4                                                         KINPUT
C                                                                       KINPUT
C     INPUT CARD E.6.B                                                  WINDOP
C                                                                       WINDOP
      READ  (5,60) (TAB(J),J=J1,J2-2),NSV,NSR                           WINDOP
```

```
60    FORMAT(3F12.0,2I12)                                               WINDOP
      TAB(J2-1) = DFLOAT(NSV)                                           WINDOP
      TAB(J2) = DFLOAT(NSR)                                             WINDOP
      IF (TAB(J1).EQ.0.0) GOTO 22                                       WINDOP
      WRITE(6,23) (TAB(J),J=J1,J2-2),NSV,SEG(NSV),NSR,SEG(NSR)          WINDOP
23    FORMAT(' SPEC. HEAT RATIO    SONIC VEL.    ABS. PRESS.',7X,       WINDOP
     *       'SEGMENT     REF. SEGMENT',/3F15.4,2(I11,A4)//)           WINDOP
      J1=J2+1                                                           WINDOP
      GOTO 30                                                           MISC
22    WRITE (6,18) (TAB(J),J=J1,J2)                                     KINPUT
   17 FORMAT(6F12.0)                                                    KINPUT
   18 FORMAT(10X,'D0',13X,'D1',13X,'D2',13X,'D3',8X,'REF. SEGMENT',     WINDOP
     *       /5F15.4//)                                                 WINDOP
      J1 = J2+1                                                         KINPUT
C                                                                       KINPUT
C     INPUT CARD E.6.C - NTMPTS                                         KINPUT
C                                                                       KINPUT
      READ  (5,11) NTMPTS                                               KINPUT
      WRITE (6,19) NTMPTS                                               KINPUT
   19 FORMAT('0 WIND FORCE TABLES FOR ',I6,' TIME POINTS.'//            KINPUT
     *       11X,'T',14X,'FX(T)',15X,'FY(T)',15X,'FZ(T)'  /)            KINPUT
      TAB(J1) = NTMPTS                                                  KINPUT
      J1 = J1+1                                                         KINPUT
      J2 = J1+4*NTMPTS-1                                                KINPUT
C                                                                       KINPUT
C     INPUT CARDS E.6.D-E.6.N - NTMPTS CARDS OF T,FX(T),FY(T),FZ(T)     KINPUT
C                                                                       KINPUT
      READ  (5,20) (TAB(J),J=J1,J2)                                     KINPUT
      WRITE (6,21) (TAB(J),J=J1,J2)                                     KINPUT
   20 FORMAT(4F12.0)                                                    KINPUT
   21 FORMAT(3X,F12.6,3G20.6)                                           KINPUT
      J1 = J2+1                                                         KINPUT
   30 CONTINUE                                                          KINPUT
   31 IF (NJNTF.LE.0) GO TO 51                                          KINPUT
      DO 50 K=1,NJNTF                                                   KINPUT
C                                                                       KINPUT
C     INPUT CARD E.7.A - FUNCTION NO. AND TITLE                         KINPUT
C                                                                       KINPUT
      READ  (5,12) I,(KTITLE(J),J=1,5)                                  KINPUT
      WRITE (6,32) I,(KTITLE(J),J=1,5),I,J1,NPG                         PAGE
      NPG=NPG+1                                                         PAGE
   32 FORMAT('1 JOINT FORCE FUNCTION NO.',I4,4X,5A4,10X,'NTI(',I2,') =',KINPUT
     *       I5,45X,'PAGE',I5/120X,'CARDS E.7'/)                        PAGE
      IF (I.LE.0.OR.I.GT.50) WRITE (6,14)                               KINPUT
      IF (I.LE.0.OR.I.GT.50) STOP 12                                    KINPUT
      IF (NTI(I).NE.0) WRITE (6,15) I                                   KINPUT
      NTI(I) = J1                                                       KINPUT
      DO 33 J=1,5                                                       KINPUT
   33 JTITLE(J,I) = KTITLE(J)                                           KINPUT
C                                                                       KINPUT
```

```
C     INPUT CARD E.7.B - D0,D1,D2,D3,D4 (FOR NOW A BLANK CARD).           KINPUT
C                                                                         KINPUT
      J2 = J1+4                                                           KINPUT
      READ  (5,17) (TAB(J),J=J1,J2)                                       KINPUT
      WRITE (6,18) (TAB(J),J=J1,J2)                                       KINPUT
      J1 = J2+1                                                           KINPUT
C                                                                         KINPUT
C     INPUT CARD E.7.C - NTHETA,NPHI                                      KINPUT
C                                                                         KINPUT
      READ  (5,11) NTHETA,NPHI                                            KINPUT
      TAB(J1  ) = NTHETA                                                  KINPUT
      TAB(J1+1) = NPHI                                                    KINPUT
      J1 = J1+2                                                           KINPUT
      IF (NTHETA.LT.0) GO TO 38                                           KINPUT
      DO 35 J=1,NTHETA                                                    KINPUT
   35 TH(J) = DFLOAT(J-1)*180.0/DFLOAT(NTHETA-1)                          KINPUT
      WRITE (6,36) NTHETA,NPHI,(TH(J),J=2,NTHETA)                         KINPUT
   36 FORMAT('0 FUNCTION IS TABULAR FOR' ,I3,' X',I3,' VALUES OF THETA AKINPUT
     *ND PHI'//30X,'THETA'/5X,'PHI',5X,'THETA0',F16.3,4F20.3/             KINPUT
     *      (15X,5F20.3))                                                 KINPUT
   37 FORMAT(F9.2,F10.3,5G20.7/(19X,5G20.7))                              KINPUT
      GO TO 40                                                            KINPUT
   38 NPOLY = -NTHETA -1                                                  KINPUT
      WRITE (6,39) NPOLY,NPHI,(BLANK,J,J=1,NPOLY)                         KINPUT
   39 FORMAT('0 FUNCTION IS COEFFICIENTS OF' ,I3,' ORDER POLYNOMIALS IN KINPUT
     *(THETA-THETA0) FOR',I3,' VALUES OF PHI.'//                          KINPUT
     *  27X,'COEFFICIENTS OF (THETA-THETA0)**N'/                          KINPUT
     *  5X,'PHI',5X,'THETA0',7X,5(A4,'N =',I2,11X)/(26X,A4,'N =',I2,11X,KINPUT
     *  A4,'N =',I2,11X,A4,'N =',I2,11X,A4,'N =',I2,11X,A4,'N =',I2)  ) KINPUT
   40 WRITE (6,21)                                                        KINPUT
      DO 49 I=1,NPHI                                                      KINPUT
      PHIDEG = DFLOAT(I-1)*360.0/DFLOAT(NPHI) - 180.0                     KINPUT
C                                                                         KINPUT
C     INPUT CARDS E.7.D - E.7.N NPHI SETS WITH NTHETA ITEMS PER SET.      KINPUT
C        EACH SET I IS FOR PHI(I) = -180 +(I-1)*360/NPHI DEGREES AND      KINPUT
C        ASSUMES DATA FOR PHI(NPHI+1) = 180 IS SAME AS PHI(1) = -180.     KINPUT
C                                                                         KINPUT
      J2 = J1 + IABS(NTHETA) -1                                           KINPUT
      READ  (5,17) (TAB(J),J=J1,J2)                                       KINPUT
      WRITE (6,37) PHIDEG,(TAB(J),J=J1,J2)                                KINPUT
      IF (NTHETA.LT.0) TAB(J1) = TAB(J1)*RADIAN                           KINPUT
      IF (NTHETA.LT.0) GO TO 49                                           KINPUT
C                                                                         KINPUT
C     FOR TABULAR DATA, FILL IN ZERO VALUES WITH INTERPOLATED NEGATIVE    KINPUT
C     VALUES. OVERWRITE VALUE IN FIRST COLUMN (SUPPLIED AS THETA0) WITH KINPUT
C     VALUE FOR THETA = 0 AND ALL OTHER ZERO VALUES.                      KINPUT
C                                                                         KINPUT
      THETA0 = TAB(J1)                                                    KINPUT
      IF (THETA0.EQ.0.0) GO TO 49                                         KINPUT
      JJ = THETA0*DFLOAT(NTHETA-1)/180.0 + 1.0 + EPS(6)                   KINPUT
```

```
      JJ1 = J1+JJ                                                     KINPUT
      IERROR = 0                                                      KINPUT
      IF (JJ1.GT.J2) IERROR = 1                                       KINPUT
      IF (TAB(JJ1).LE.0.0) IERROR = 2                                 KINPUT
      IF (IERROR.NE.0) GO TO 46                                       KINPUT
      DO 45 J=1,JJ                                                    KINPUT
      J1J = J1+J-1                                                    KINPUT
      IF (J.NE.1.AND.TAB(J1J).GT.0.0) IERROR = 3                      KINPUT
   45 TAB(J1J) = TAB(JJ1)*(TH(J)-THETA0)/(TH(JJ+1)-THETA0)            KINPUT
   46 IF (IERROR.NE.0) WRITE (6,47) IERROR                            KINPUT
   47 FORMAT('0 INPUT ERROR. INCONSISTENT VALUE OF THETA0. IERROR =',I2,KINPUT
     *        ' PROGRAM TERMINATED.')                                 KINPUT
      IF (IERROR.NE.0) STOP 13                                        KINPUT
   49 J1 = J2+1                                                       KINPUT
   50 CONTINUE                                                        KINPUT
   51 MXTB1 = J1-1                                                    KINPUT
      RETURN                                                          KINPUT
      END                                                             KINPUT
```

```
      SUBROUTINE LINAXS(X0,Y0,THETA,NINTVS,TOTLGT)                      LINAXS
C                                                     REV 18    02/28/78LINAXS
C     PURPOSE : PREPARE A LINEAR AXIS ON A PLOT.                        LINAXS
C                                                                       LINAXS
C     DESCRIPTION OF PARAMETERS:                                        LINAXS
C        X0,Y0 - STARTING POINT (IN INCHES, REL TO PLOTTER ORIGIN).     LINAXS
C                                                                       LINAXS
C        THETA - ANGLE OF AXIS, IN DEGREES.                             LINAXS
C                                                                       LINAXS
C        NINTVS- MAGNITUDE = NO. OF INTERVALS DELINEATED BY TIC MARKS.  LINAXS
C              - SIGN DETERMINES WHETHER TIC MARKS ARE PLACED ON        LINAXS
C                POSITIVE OR NEGATIVE SIDE OF AXIS, RESPECTIVELY        LINAXS
C                (POSITIVE SIDE IS TO LEFT OF DIRECTION OF TRAVEL).     LINAXS
C                                                                       LINAXS
C        TOTLGT- TOTAL LENGTH OF AXIS, IN INCHES.                       LINAXS
C                                                                       LINAXS
C     SUBROUTINES REQUIRED : SIN, COS, PLOT (NOTE: SINGLE PRECISION).   LINAXS
C                                                                       LINAXS
C     AUTHOR: W. D. FRYER, CALSPAN (MARCH 1967).                        LINAXS
C                                                                       LINAXS
C     PLAGIARIZED FROM CALSPAN SUBROUTINE LIBRARY (NO. CU 0035).        LINAXS
C                                                                       LINAXS
      THR = 1.7453293E-2 * THETA                                        LINAXS
      SINT = SIN(THR)                                                   LINAXS
      COST = COS(THR)                                                   LINAXS
C                                                                       LINAXS
      DL = ABS(TOTLGT/ FLOAT(NINTVS))                                   LINAXS
      DX = DL*COST                                                      LINAXS
      DY = DL*SINT                                                      LINAXS
C                                                                       LINAXS
      TICX = -0.12* SINT                                                LINAXS
      TICY = 0.12* COST                                                 LINAXS
      IF(NINTVS.GT.0) GO TO 30                                          LINAXS
      TICX = -TICX                                                      LINAXS
      TICY = -TICY                                                      LINAXS
C                                                                       LINAXS
   30 X = X0                                                            LINAXS
      Y = Y0                                                            LINAXS
C                                                                       LINAXS
      CALL PLOT (X +TICX,Y+TICY,3)                                      LINAXS
      CALL PLOT (X,Y,2)                                                 LINAXS
      NINT = IABS(NINTVS)                                               LINAXS
      DO 40 I=1,NINT                                                    LINAXS
      X = X+DX                                                          LINAXS
      Y = Y+DY                                                          LINAXS
      CALL PLOT(X,Y,2)                                                  LINAXS
      CALL PLOT(X+TICX,Y+TICY,2)                                        LINAXS
   40 CALL PLOT(X,Y,2)                                                  LINAXS
C                                                                       LINAXS
      RETURN                                                            LINAXS
      END                                                               LINAXS
```

```
      SUBROUTINE LOGAXS(XO,YO,THETA,NDEC,EXTENT)                        LOGAXS
C                                                 REV 19    09/18/79LOGAXS
C     PURPOSE : PREPARE LOGARITHMIC AXIS ON A PLOT.                     LOGAXS
C                                                                       LOGAXS
C     DESCRIPTION OF PARAMETERS:                                        LOGAXS
C                                                                       LOGAXS
C        XO,YO - STARTING POINT (IN INCHES, REL TO PLOTTER ORIGIN).     LOGAXS
C                                                                       LOGAXS
C        THETA - ANGLE OF AXIS (DEGREES).                               LOGAXS
C                                                                       LOGAXS
C        NDECS - MAGNITUDE OF NDECS SPECIFIES NO. OF DECADES.           LOGAXS
C              - SIGN DETERMINES WHETHER TIC MARKS ARE TO BE PLACED     LOGAXS
C                ON POS. OR NEG. SIDE OF AXIS, RESP. (POS. SIDE IS      LOGAXS
C                TO LEFT OF PREDOMINANT DIRECTION OF TRAVEL).           LOGAXS
C                                                                       LOGAXS
C        EXTENT- MAGNITUDE OF EXTENT SETS OVER-ALL LENGTH OF AXIS       LOGAXS
C                IN INCHES. IF EXTENT IS POSITIVE, TIC MARKS ARE        LOGAXS
C                SPACED NORMALLY (LARGE INTERVALS FIRST). IF EXTENT     LOGAXS
C                IS NEGATIVE, TIC MARKS ARE SPACED IN REVERSE ORDER     LOGAXS
C                (SMALL INTERVALS FIRST).                               LOGAXS
C                                                                       LOGAXS
C     SUBROUTINES REQUIRED : SIN, COS, PLOT (NOTE: SINGLE PRECISION).   LOGAXS
C                                                                       LOGAXS
C     AUTHOR: W. D. FRYER, CALSPAN (MARCH 1967).                        LOGAXS
C                                                                       LOGAXS
C     PLAGIARIZED FROM CALSPAN SUBROUTINE LIBRARY (NO. CU 0036).        LOGAXS
C                                                                       LOGAXS
      LOGICAL REVERS                                                    LOGAXS
      REAL XL(18),XLO(19)                                               LOGAXS
      EQUIVALENCE (XLO(2),XL(1))                                        LOGAXS
      DATA XLO/ 0.0    , 0.17609, 0.30103, 0.39794, 0.47712, 0.54407,   LOGAXS
     * 0.60206, 0.65321, 0.69897, 0.74036, 0.77815, 0.81291, 0.84510,   LOGAXS
     * 0.87506, 0.90309, 0.92942, 0.95424, 0.97772, 1.0     /           LOGAXS
      DATA RPD /1.7453293E-2/                                           LOGAXS
C                                                                       LOGAXS
C                                                                       LOGAXS
      REVERS = .FALSE.                                                  LOGAXS
      IF(EXTENT.LT.0.0) REVERS = .TRUE.                                 LOGAXS
C                                                                       LOGAXS
      NODEC = IABS(NDEC)                                                LOGAXS
      SPDEC = ABS(EXTENT) / FLOAT(NODEC)                                LOGAXS
      THR = THETA*RPD                                                   LOGAXS
      COST = COS(THR)                                                   LOGAXS
      SINT = SIN(THR)                                                   LOGAXS
C                                                                       LOGAXS
      TICX1 =-0.05*SINT                                                 LOGAXS
      TICY1 = 0.05*COST                                                 LOGAXS
      TICXA = -0.12*SINT                                                LOGAXS
      TICXB = -0.20*SINT                                                LOGAXS
```

```
      TICYA = 0.12*COST                                                 LOGAXS
      TICYB = 0.20*COST                                                 LOGAXS
      IF(NDEC.GT.0) GO TO 50                                            LOGAXS
C                                                                       LOGAXS
      TICX1 = -TICX1                                                    LOGAXS
      TICY1 = - TICY1                                                   LOGAXS
      TICX2 = -TICX2                                                    LOGAXS
      TICXA = - TICXA                                                   LOGAXS
      TICYA = -TICYA                                                    LOGAXS
      TICXB = -TICXB                                                    LOGAXS
      TICYB = - TICYB                                                   LOGAXS
C                                                                       LOGAXS
   50 COST = COST*SPDEC                                                 LOGAXS
      SINT = SINT* SPDEC                                                LOGAXS
      TICX2 = TICXA                                                     LOGAXS
      TICY2 = TICYA                                                     LOGAXS
C                                                                       LOGAXS
      XD = X0                                                           LOGAXS
      YD = Y0                                                           LOGAXS
      ND = 1                                                            LOGAXS
      N = 0                                                             LOGAXS
C                                                                       LOGAXS
C     *****GO TO START POS.*****                                        LOGAXS
      CALL PLOT(X0+TICXB,Y0+TICYB,3)                                    LOGAXS
      CALL PLOT(X0,Y0,2)                                                LOGAXS
C                                                                       LOGAXS
   60 N = N+1                                                           LOGAXS
      Q = XL(N)                                                         LOGAXS
      IF(.NOT. REVERS) GO TO 65                                         LOGAXS
      M = 18-N                                                          LOGAXS
      Q = 1.0-XL(M)                                                     LOGAXS
   65 X = XD + Q*COST                                                   LOGAXS
      Y = YD + Q*SINT                                                   LOGAXS
      CALL PLOT(X,Y,2)                                                  LOGAXS
      CALL PLOT(X+TICX1,Y+TICY1,2)                                      LOGAXS
      CALL PLOT (X,Y,2 )                                                LOGAXS
C                                                                       LOGAXS
      N = N+1                                                           LOGAXS
      Q = XL(N)                                                         LOGAXS
      IF(.NOT. REVERS) GO TO 75                                         LOGAXS
      M = 18-N                                                          LOGAXS
      Q = 1.0 - XL(M)                                                   LOGAXS
   75 X = XD + Q*COST                                                   LOGAXS
      Y = YD + Q*SINT                                                   LOGAXS
      CALL PLOT(X,Y,2)                                                  LOGAXS
      CALL PLOT (X+TICX2,Y+TICY2,2)                                     LOGAXS
      CALL PLOT(X,Y,2)                                                  LOGAXS
C                                                                       LOGAXS
      IF(N-16) 60,80,100                                                LOGAXS
C                                                                       LOGAXS
```

```
   80 TICX2 = TICXB                                                    LOGAXS
      TICY2 = TICYB                                                    LOGAXS
      GO TO 60                                                         LOGAXS
C                                                                      LOGAXS
  100 IF(ND .EQ. NODEC) GO TO 200                                      LOGAXS
      TICX2 = TICXA                                                    LOGAXS
      TICY2 = TICYA                                                    LOGAXS
      N = 0                                                            LOGAXS
      XD = X                                                           LOGAXS
      YD = Y                                                           LOGAXS
      ND = ND+1                                                        LOGAXS
      GO TO 60                                                         LOGAXS
C                                                                      LOGAXS
  200 RETURN                                                           LOGAXS
      END                                                              LOGAXS
```

```
      FUNCTION LTIME(N)                                                 LTIME
C                                                REV III.2 08/08/84REVIII
C     TEMPORARY FORTRAN VERSION OF S/370 ASSEMBLER LANGUAGE ROUTINE FROMLTIME
C     CALSPAN LIBRARY THAT MEASURES ELAPSED CPU TIME IN UNITS OF 0.01   LTIME
C     SECONDS. IT SHOULD BE REPLACED WITH AN EQUIVALENT ROUTINE BY THE  LTIME
C     USER TO ENABLE SUBROUTINE ELTIME TO PERFORM ON HIS COMPUTER.      LTIME
C                                                                       LTIME
C     ORIGINAL CALSPAN ROUTINE PERFORMS AS FOLLOWS:                     LTIME
C         IT = LTIME(0) GIVES ELAPSED CPU TIME (INTEGER NUMBER OF 0.01  LTIME
C                       SECOND UNITS) SINCE SUBROUTINE REFERENCE WAS    LTIME
C                       RESET, AND RESETS THIS REFERENCE.               LTIME
C         IT = LTIME(1) SAME, EXCEPT THAT THE REFERENCE IS NOT RESET.   LTIME
C                                                                       LTIME
C                                                                       PECONV
C     THIS SUBROUTINE DOESN'T WORK WITH THE P-E COMPUTER                PECONV
C     BUT THE CODE IS LEFT HERE AS A DUMMY SUBROUTINE.                  PECONV
C     HOWEVER, THERE IS A VERSION OF THIS SUBROUTINE THAT               PECONV
C     CAN BE USED, BUT IT CAN ONLY BE COMPILED WITH THE                 PECONV
C     P-E FORTRAN O COMPILER.  THE OBJECT DECK FOR THIS                 PECONV
C     SUBROUTINE IS KEPT SEPARATELY AND INCLUDED IN THE                 PECONV
C     TASK FILE WHEN THE PROGRAM IS LINKED                              PECONV
C                                                                       PECONV
      DATA KTIME/0/                                                     LTIME
      KTIME = KTIME+1                                                   LTIME
      LTIME = KTIME                                                     LTIME
      IF (N.EQ.0) KTIME = 0                                             LTIME
      RETURN                                                            LTIME
      END                                                               LTIME
```

```
      SUBROUTINE MAT31 (A,B,C)                                          MAT31
C                                                   REV 17    01/03/77MAT31
C     PERFORMS MATRIX MULTIPLICATION C = AB                             MAT31
C     WHERE A IS A 3X3 MATRIX, AND B AND C ARE VECTORS OF LENGTH 3.     MAT31
C                                                                       MAT31
      IMPLICIT REAL*8 (A-H,O-Z)                                         MAT31
      DIMENSION  A(3,3) , B(3) , C(3)                                   MAT31
      C(1) = A(1,1)*B(1) + A(1,2)*B(2) + A(1,3)*B(3)                    MAT31
      C(2) = A(2,1)*B(1) + A(2,2)*B(2) + A(2,3)*B(3)                    MAT31
      C(3) = A(3,1)*B(1) + A(3,2)*B(2) + A(3,3)*B(3)                    MAT31
      RETURN                                                            MAT31
      END                                                               MAT31
```

```
      SUBROUTINE MAT33 (A,B,C)                                          MAT33
C                                                       REV 17    01/03/77MAT33
C     PERFORMS MATRIX MULTIPLICATION C = AB                             MAT33
C     WHERE A, B AND C ARE ALL 3X3 MATRICEES.                           MAT33
C                                                                       MAT33
      IMPLICIT REAL*8 (A-H,O-Z)                                         MAT33
      DIMENSION  A(3,3) , B(3,3) , C(3,3)                               MAT33
      DO  10  I=1,3                                                     MAT33
      DO  10  J=1,3                                                     MAT33
   10 C(I,J) = A(I,1)*B(1,J) + A(I,2)*B(2,J) + A(I,3)*B(3,J)            MAT33
      RETURN                                                            MAT33
      END                                                               MAT33
```

```
      SUBROUTINE ORTHO(P,X,L)                                           ORTHO
C                                                 REV 03    05/31/73ORTHO
C     GENERATES A SET OF RIGHT HANDED ORTHONORMAL VECTORS (P),          ORTHO
C     GIVEN ONE OF THE VECTORS (X), WHERE                               ORTHO
C          P - LX3 MATRIX OF 3 ORTHONORMAL VECTORS TO BE GENERATED.     ORTHO
C          X - GIVEN VECTOR.                                            ORTHO
C          L - 1ST SUBSCRIPT OF P IN CALLING PROGRAM.                   ORTHO
C                                                                       ORTHO
      IMPLICIT REAL*8(A-H,O-Z)                                          ORTHO
      DIMENSION P(L,3),X(3)                                             ORTHO
       M=2                                                              ORTHO
       N=3                                                              ORTHO
       TEST=0.                                                          ORTHO
       DO 5 I=1,3                                                       ORTHO
       P(I,3)=X(I)                                                      ORTHO
       D=1.-X(I)**2                                                     ORTHO
       IF(D.LE.TEST)GO TO 4                                             ORTHO
       TEST=D                                                           ORTHO
       D=DSQRT(D)                                                       ORTHO
       P(I,1)=D                                                         ORTHO
       P(I,2)=0.                                                        ORTHO
       P(M,2)=X(N)/D                                                    ORTHO
       P(N,2)=-X(M)/D                                                   ORTHO
       P(M,1)=X(I)*P(N,2)                                               ORTHO
       P(N,1)=-X(I)*P(M,2)                                              ORTHO
    4  M=N                                                              ORTHO
       N=I                                                              ORTHO
    5  CONTINUE                                                         ORTHO
       RETURN                                                           ORTHO
      END                                                               ORTHO
```

```
      SUBROUTINE OUTPUT(IJK)                                            OUTPUT
C                                                   REV IV   02/01/88MISDOT
C     CONTROLS TABULATED OUTPUT ON FORTRAN UNITS (STARTING WITH NO. 21) OUTPUT
C     OF SELECTED OPTIONAL SEGMENT LINEAR AND ANGULAR ACCELERATIONS,    OUTPUT
C     VELOCITIES AND DISPLACEMENTS, JOINT PARAMETERS AND SELECTED DATA  OUTPUT
C     FROM ALL ALLOWED CONTACT FORCE COMPUTATIONS BETWEEN BODY SEGMENTS OUTPUT
C     AND VEHICLE COMPONENTS.                                           OUTPUT
C                                                                       OUTPUT
      IMPLICIT REAL*8 (A-H,O-Z)                                         OUTPUT
      COMMON/CONTRL/ TIME,NSEG,NJNT,NPL,NBLT,NBAG,NVEH,NGRND,           OUTPUT
     *               NS,NQ,NSD,NFLX,NHRNSS,NWINDF,NJNTF,NPRT(36),NPG    PAGE
      COMMON/CMATRX/ V1(3,30),V2(3,30),V3(3,12),B12(3,3,60),A22(3,3,60),OUTPUT
     *               F(3,30),TQ(3,30),WJ(30),A11(3,3,30)                SLIP
      COMMON/SGMNTS/ D(3,3,30),WMEG(3,30),WMEGD(3,30),U1(3,30),U2(3,30),OUTPUT
     *               SEGLP(3,30),SEGLV(3,30),SEGLA(3,30),NSYM(30)       OUTPUT
      COMMON/DESCRP/ PHI(3,30),W(30),RW(30),SR(4,60),HA(3,60),HB(3,60), SLIP
     *               RPHI(3,30),HT(3,3,60),SPRING(5,90),VISC(7,90),     OUTPUT
     *               JNT(30),IPIN(30),ISING(30),IGLOB(30),JOINTF(30)    OUTPUT
      COMMON/JBARTZ/ MNPL(   30),MNBLT(   8),MNSEG(   30),MNBAG(    6), OUTPUT
     *               MPL(3,5,30),MBLT(3,5,8),MSEG(3,5,30),MBAG(3,10,6), OUTPUT
     *               NTPL( 5,30),NTBLT( 5,8),NTSEG( 5,30)               OUTPUT
      COMMON/TITLES/ DATE(3),COMENT(40),VPSTTL(20),BDYTTL(5),           OUTPUT
     *               BLTTTL(5,8),PLTTL(5,30),BAGTTL(5,6),SEG(30),       OUTPUT
     *               JOINT(30),CGS(30),JS(30)                           OUTPUT
      REAL DATE,COMENT,VPSTTL,BDYTTL,BLTTTL,PLTTL,BAGTTL,SEG,JOINT      OUTPUT
      LOGICAL*1 CGS,JS                                                  OUTPUT
      COMMON/FORCES/PSF(7,70),BSF(4,20),SSF(10,40),BAGSF(3,20),         NCFORC
     *               PRJNT(7,30),NPANEL(5),NPSF,NBST,NSSF,NBGSF         OUTPUT
      COMMON/CNSNTS/ PI,RADIAN,G,THIRD,EPS(24),                         OUTPUT
     *               UNITL,UNITM,UNITT,GRAVTY(3),TWOPI                  TWOPI
      COMMON/RSAVE/  XSG(3,20,3),DPMI(3,3,30),LPMI(30),                 ATBIII
     *               NSG(9),MSG(20,9),MCG,MCGIN(24,5),KREF(20,9)        TTHKREF
      COMMON/COMAIN/VAR(240),DER(240),DT,H0,HMAX,HMIN,RSTIME,           OUTPUT
     *              ISTEP,NSTEPS,NDINT,NEQ,IRSIN,IRSOUT                 OUTPUT
      COMMON/DAMPER/ APSDM(3,20),APSDN(3,20),ASD(5,20),MSDM(20),MSDN(20)OUTPUT
      COMMON/HRNESS/ BAR(15,100),BB(100),BBDOT(100),PLOSS(2,100),       OUTPUT
     *               XLONG(20),HTIME(2),IBAR(5,100),NL(2,100),          OUTPUT
     *               NPTSPB(20),NPTPLY(20),NTHRNS(20),NBLTPH(5)         OUTPUT
      COMMON/WINDFR/ WTIME(30),QFU(3,5),QFV(3,5),WF(3,30),IWIND(30),    WINDOP
     *               MWSEG(7,30),NFVSEG(6),NFVNT(5),MOWSEG(30,30)       WINDOP
      COMMON/TEMPVS/ TDATA(14,65),ACC(7,20),T1(3),T2(3),T3(3),T4(9)     CHGIII
     *               ,T5(3,3),T6(3,3),T7(3)                             CHGIII
      LOGICAL  LTAPE8 , LTHIST                                          OUTPUT
      DATA LINES/-1/,LPP/45/,NTMAX/65/                                  CHGIII
      DATA KMAX/20/,NMAX/22/,MCGMAX/5/                                  CHGIII
C                                                                       OUTPUT
      IF (IJK.NE.0) GO TO 13                                            OUTPUT
C                                                                       OUTPUT
C     SET ALL FORCE ARRAYS TO ZERO.                                     OUTPUT
C                                                                       OUTPUT
```

```
      DO 2 I=1,7                                                         MISDOT
      DO 2 J=1,70                                                        MISDOT
2     PSF(I,J) = 0.0                                                     MISDOT
      DO 3 I=1,4                                                         MISDOT
      DO 3 J=1,20                                                        MISDOT
3     BSF(I,J) = 0.0                                                     MISDOT
      DO 4 I=1,10                                                        MISDOT
      DO 4 J=1,40                                                        MISDOT
4     SSF(I,J) = 0.0                                                     MISDOT
      DO 5 I=1,3                                                         MISDOT
      DO 5 J=1,20                                                        MISDOT
5     BAGSF(I,J) =0.0                                                    MISDOT
      DO 6 I=1,7                                                         MISDOT
      DO 6 J=1,30                                                        MISDOT
6     PRJNT(I,J) = 0.0                                                   MISDOT
      GO TO 66                                                           OUTPUT
C                                                                        OUTPUT
C     LTHIST = TRUE MEANS PRINT LINE OF TIME HISTORY DATA FOR THIS       OUTPUT
C                         TIME POINT ON EACH OUTPUT UNIT (NT).           OUTPUT
C                                                                        OUTPUT
C     LTAPE8 = TRUE MEANS WRITE TIME HISTORY DATA ON TAPE 8.             OUTPUT
C                                                                        OUTPUT
   13 NPRT4 = NPRT(4) + 4                                                OUTPUT
      IF  (NPRT4.LE.0 .OR. NPRT4.GT.8)  STOP 37                          OUTPUT
      IF(NPRT(26).EQ.6) GO TO 66                                         TGMOD1
      GO TO  (66,66,66,15,16,17,17,16) , NPRT4                           OUTPUT
   15 LTAPE8 = .FALSE.                                                   OUTSTP
      LTHIST = .TRUE.                                                    TGMOD1
      GO TO 116                                                          TGMOD1
   16 LTHIST = .TRUE.                                                    TGMOD1
      LTAPE8 = .TRUE.                                                    TGMOD1
      GO TO 116                                                          TGMOD1
   17 LTHIST = .FALSE.                                                   TGMOD1
      LTAPE8 = .TRUE.                                                    TGMOD1
      GO TO 217                                                          TGMOD1
  116 TEST = DMOD(TIME,DT)                                               OUTSTP
      TEST = DMIN1(TEST,DABS(DT-TEST))                                   OUTSTP
      IF ((NPRT(26).EQ.0.OR.NPRT(26).EQ.3).AND.TEST.GE.EPS(8))           TGMOD1
     * LTHIST=.FALSE.                                                    TGMOD1
      IF(.NOT.LTAPE8.AND..NOT.LTHIST) GO TO 66                           FIXTTH
  217 CONTINUE                                                           TGMOD1
      IF(NPRT(26).EQ.4) LTHIST=.FALSE.                                   TGMOD1
      IF(NPRT(26).EQ.5) LTAPE8=.FALSE.                                   TGMOD1
      IF(.NOT.LTAPE8.AND..NOT.LTHIST) GO TO 66                           TGMOD1
      CALL ELTIME (1,8)                                                  OUTPUT
      IF (LINES.GE.0) GO TO 21                                           FIXTTH
      PREVT = -999.0                                                     OUTPUT
      LINES = 0                                                          FIXTTH
      IF  (IRSIN.NE.0)  GO TO 10                                         OUTPUT
C                                                                        OUTPUT
```

```
C     1ST TIME IN ROUTINE, READ CARD INPUT FOR OUTPUT CONTROL.          OUTPUT
C                                                                       OUTPUT
C      1. NO. OF POINT TOTAL ACCELERATIONS ,POINT NOS. AND LOCATIO      CHGIII
C      2. NO. OF POINT REL. VELOCITIES    ,POINT NOS. AND LOCATION      CHGIII
C      3. NO. OF POINT REL. LINEAR  DISPLACEMENTS ,POINT NOS. AND LOCATICHGIII
C      4. NO. OF SEGMENT ANGULAR ACCELERATIONS AND SEGMENT NOS.         CHGIII
C      5. NO. OF SEGMENT REL. ANGULAR VELOCITIES    AND SEGMENT NOS.    CHGIII
C      6. NO. OF SEGMENT REL. ANGULAR DISPLACEMENTS AND SEGMENT NOS.    CHGIII
C      7. NO. OF JOINT PARAMETERS AND JOINT NOS.                        OUTPUT
C      8. NO. OF SEGMENT WIND FORCES AND SEGMENT NOS.                   WINDOP
C      9. NO. OF JOINT FORCES AND TORQUE NOS.                           WINDOP
C     10. NO. OF CENTER OF GRAVITY AND RELATED INFORMATION              WINDOP
C                                                                       OUTPUT
      WRITE(6,478)                                                      CHGIII
  478 FORMAT(1X,/,2X,'TABULAR TIME HISTORY CONTROL PARAMETERS')         CHGIII
      WRITE(6,479)                                                      CHGIII
  479 FORMAT(3X,'TYPE  KSG     SELECTED SEGMENTS OR JOINTS')            TTHKREF
      DO 20 K=1,9                                                       WINDOP
C                                                                       OUTPUT
C     INPUT CARDS H.(K).(J) FOR K=1,3                                   OUTPUT
C                                                                       OUTPUT
      IF (K.LE.3) READ (5,18) KSG,KREF(1,K),MSG(1,K),(XSG(I,1,K),I=1,3) TTHKREF
   18 FORMAT (I6,2I3,3F12.6)                                            TTHKREF
      IF (KSG.GT.KMAX) STOP 84                                          CHGIII
      IF (K.GT.3) GO TO 201                                             ATBIII
      IF (KSG.LE.1) READ(5,213) IDUMMY                                  ATBIII
  213 FORMAT(I2)                                                        ATBIII
      IF (KSG.LE.1) GO TO 201                                           ATBIII
      DO 205 J=2,KSG                                                    ATBIII
      READ (5,210) KREF(J,K),MSG(J,K),(XSG(I,J,K),I=1,3)                TTHKREF
  210 FORMAT (I9,I3,3F12.6)                                             TTHKREF
  205 CONTINUE                                                          ATBIII
  201 CONTINUE                                                          ATBIII
C                                                                       OUTPUT
C     INPUT CARDS H.(K)      FOR K=4,9                                  WINDOP
C                                                                       OUTPUT
      IF (K.GT.3) READ (5,19) KSG,(KREF(J,K),MSG(J,K),J=1,KSG)          TTHKREF
   19 FORMAT(I6,22I3/(I9,21I3))                                         TTHKREF
      IF (KSG.GT.KMAX) STOP 85                                          CHGIII
      WRITE (6,78) K,KSG,(MSG(J,K),J=1,KSG)                             TTHKREF
      WRITE (6,81) (KREF(J,K),J=1,KSG)                                  TTHKREF
   78 FORMAT('    H.',I1,1X,I3,3X,20I3)                                 TTHKREF
   81 FORMAT('         REF    ',20I3)                                   TTHKREF
      DO 80 J=1,KSG                                                     TTHKREF
      IF(KREF(J,K).GT.NGRND.OR.KREF(J,K).LT.0) STOP 55                  TTHKREF
   80 CONTINUE                                                          TTHKREF
      IF (K.NE.7 .OR. KSG.EQ.0) GO TO 20                                OUTPUT
      DO 12 J=1,KSG                                                     OUTPUT
      L = MSG(J,K)                                                      OUTPUT
      IF (IABS(IPIN(L)).EQ.4) MSG(J,K) = -L                             OUTPUT
```

```
   12 CONTINUE                                                          OUTPUT
   20 NSG(K) = KSG                                                      OUTPUT
C                                                                       ATBIII
C     READ INPUT CARDS H.10                                             WINDOP
C                                                                       ATBIII
      READ (5,111) MCG                                                  ATBIII
  111 FORMAT(I6)                                                        ATBIII
      IF (MCG.GT.MCGMAX) STOP 86                                        CHGIII
      IF (MCG.EQ.0) GO TO 114                                           ATBIII
      DO 113 K=1,MCG                                                    ATBIII
      READ (5,112) M,N,(MCGIN(I+2,K),I=1,N)                             ATBIII
  112 FORMAT (24I3)                                                     ATBIII
      IF (N.GT.NMAX) STOP 87                                            CHGIII
      WRITE (6,117) N,(MCGIN(I+2,K),I=1,N)                              TTHKREF
  117 FORMAT('    H.10',I3,3X,22I3)                                     TTHKREF
      WRITE (6,81) M                                                    TTHKREF
      MCGIN(1,K) = M                                                    ATBIII
  113 MCGIN(2,K) = N                                                    ATBIII
  114 CONTINUE                                                          ATBIII
   10 IF  (.NOT.LTAPE8)  GO TO 21                                       OUTPUT
      WRITE  (8)         NSEG,NJNT,NPL,NBLT,NBAG,NVEH,NGRND,NPANEL,     OUTPUT
     *                   MNPL,MNBLT,MNSEG,MNBAG,MPL,MBLT,MSEG,MBAG      OUTPUT
      WRITE  (8)         DATE,COMENT,VPSTTL,BDYTTL,BLTTTL,PLTTL,BAGTTL, OUTPUT
     *                   SEG,JOINT,UNITL,UNITM,UNITT,NSG,MSG,XSG,MCG,   ATBIII
     *                   MCGIN,KREF,NHRNSS,NBLTPH,NPTSPB,NSD,MSDM,MSDN  CHGIII
   21 IF(LTHIST) LINES= LINES + 1                                       FIXTTH
      IF  (MOD(LINES,LPP).EQ.1 .AND. LTHIST)  CALL HEDING (LINES,LPP)   OUTPUT
      NT = 20                                                           OUTPUT
      USEC = 1000.0*TIME                                                OUTPUT
C                                                                       OUTPUT
C     COMPUTE AND PRINT DATA FOR 9 TYPES OF OUTPUT ABOVE                WINDOP
C                                                                       OUTPUT
      DO 44 K=1,9                                                       WINDOP
      IF (NSG(K).LE.0) GO TO 44                                         OUTPUT
      KSG = NSG(K)                                                      OUTPUT
      IF (K.GT.8) GO TO 440                                             WINDOP
      J3  = 3                                                           OUTPUT
      IF (K.EQ.7) J3 = 2                                                OUTPUT
      DO 43 J1=1,KSG,J3                                                 OUTPUT
      J2 = MIN0(J1+J3-1,KSG)                                            OUTPUT
      NT =NT + 1                                                        OUTPUT
C     SETUP LOGICAL UNIT CONTROL (FOR PRINTER) FOR PERKIN & ELMER       PECONV
      CALL CARCON(NT,1)                                                 PECONV
      DO 38 J=J1,J2                                                     OUTPUT
      L = IABS(MSG(J,K))                                                OUTPUT
      GO TO (22,24,26,29,31,34,35,601),K                                WINDOP
C                                                                       OUTPUT
C     1. POINT TOTAL ACCELERATION IN KREF(1) REFERENCE                  CHGIII
C                                                                       OUTPUT
   22 IF(LPMI(L).EQ.0) GO TO 521                                        CHGIII
```

```
      CALL MAT31(DPMI(1,1,L),XSG(1,J,K),T7)                            CHGIII
      GO TO 523                                                        CHGIII
  521 DO 522 JL=1,3                                                    CHGIII
  522 T7(JL) = XSG(JL,J,K)                                             CHGIII
  523 CALL CROSS (WMEG(1,L),T7,T1)                                     CHGIII
      CALL CROSS (WMEG(1,L),T1,T2)                                     OUTPUT
      CALL CROSS (WMEGD(1,L),T7,T3)                                    CHGIII
      CALL MAT31(D(1,1,L),GRAVTY,T7)                                   ACCEL
      CALL MAT31(D(1,1,L),SEGLA(1,L),T4)                               OUTPUT
      DO 23 I=1,3                                                      OUTPUT
      IF(MSG(J,K).LT.0) T4(I)=T4(I)+T7(I)                              ACCEL
      ACC(I,J) = (T4(I)+T3(I)+T2(I))/G                                 OUTPUT
   23 T1(I) = ACC(I,J)                                                 OUTPUT
      IF(MSG(J,K).GE.0) GO TO 405                                      ACCEL
      KRF=L                                                            ACCEL
      IF(LPMI(KRF).NE.0) CALL DOT31(DPMI(1,1,KRF),T1,ACC(1,J))         ACCEL
      IF(KREF(J,K).EQ.1) GOTO 33                                       ACCEL
      DO 600 II=1,3                                                    ACCEL
  600 ACC(II,J)=ACC(II,J)-GRAVTY(II)/G                                 ACCEL
      GOTO 33                                                          ACCEL
C                                                                      OUTPUT
C     2. POINT REL. VELOCITY IN KREF(2) REFERENCE                      CHGIII
C                                                                      OUTPUT
   24 IF(KREF(J,2).EQ.0) KRF = NVEH                                    TTHKREF
      IF(KREF(J,2).NE.0) KRF = KREF(J,2)                               TTHKREF
      IF(LPMI(L).EQ.0) GO TO 524                                       CHGIII
      CALL MAT31(DPMI(1,1,L),XSG(1,J,K),T7)                            CHGIII
      GO TO 525                                                        CHGIII
  524 DO 526 JL=1,3                                                    CHGIII
  526 T7(JL) = XSG(JL,J,K)                                             CHGIII
  525 CALL CROSS (WMEG(1,L),T7,T1)                                     CHGIII
      CALL DOT31(D(1,1,L),T1,T2)                                       OUTPUT
      DO 25 I=1,3                                                      OUTPUT
   25 T3(I) = T2(I) + SEGLV(I,L) - SEGLV(I,KRF)                        CHGIII
      GO TO 28                                                         OUTPUT
C                                                                      OUTPUT
C     3. POINT REL. LINEAR DISPLACEMENT IN KREF(3) REFERENCE           CHGIII
C                                                                      OUTPUT
   26 IF(KREF(J,3).EQ.0) KRF = NVEH                                    TTHKREF
      IF(KREF(J,3).NE.0) KRF = KREF(J,3)                               TTHKREF
      IF  (LPMI(L).EQ.0)  GO TO 76                                     CHGIII
      CALL DOT33 (DPMI(1,1,L),D(1,1,L),T4)                             OUTPUT
      CALL DOT31 (T4,XSG(1,J,K),T1)                                    OUTPUT
      GO TO 77                                                         OUTPUT
   76 CALL DOT31 (D(1,1,L),XSG(1,J,K),T1)                              OUTPUT
   77 DO 27 I=1,3                                                      OUTPUT
   27 T3(I) = T1(I) + SEGLP(I,L) - SEGLP(I,KRF)                        CHGIII
   28 IF (LPMI(KRF).EQ.0) GO TO 403                                    CHGIII
      CALL DOT33(DPMI(1,1,KRF),D(1,1,KRF),T5)                          CHGIII
      CALL MAT31(T5,T3,ACC(1,J))                                       CHGIII
```

```
      GO TO 33                                                          OUTPUT
  403 CALL MAT31(D(1,1,KRF),T3,ACC(1,J))                                CHGIII
   33 ACC(4,J) = DSQRT(ACC(1,J)**2+ACC(2,J)**2+ACC(3,J)**2)             CHGIII
      GO TO 38                                                          CHGIII
C                                                                       OUTPUT
C     4. SEGMENT ANGULAR ACCELERATION IN KREF(4) REFERENCE              CHGIII
C                                                                       OUTPUT
   29 DO 30 I=1,3                                                       OUTPUT
      ACC(I,J) = WMEGD(I,L)/(2.0*PI)                                    OUTPUT
   30 T1(I) = ACC(I,J)                                                  OUTPUT
  405 CONTINUE                                                          CHGIII
      IF(KREF(J,K).EQ.0) GO TO 401                                      TTHKREF
      KRF = KREF(J,K)                                                   TTHKREF
      IF(LPMI(KRF).EQ.0) GO TO 402                                      CHGIII
      CALL DOT33(DPMI(1,1,KRF),D(1,1,KRF),T5)                           CHGIII
      CALL DOTT33(T5,D(1,1,L),T6)                                       CHGIII
      CALL MAT31(T6,T1,ACC(1,J))                                        CHGIII
      GO TO 33                                                          CHGIII
  402 CALL DOTT33(D(1,1,KRF),D(1,1,L),T6)                               CHGIII
      CALL MAT31(T6,T1,ACC(1,J))                                        CHGIII
      GO TO 33                                                          CHGIII
  401 KRF = L                                                           CHGIII
      IF(LPMI(KRF).NE.0) CALL DOT31(DPMI(1,1,KRF),T1,ACC(1,J))          CHGIII
      GO TO 33                                                          OUTPUT
C                                                                       OUTPUT
C     5. SEGMENT REL. ANGULAR VELOCITY IN KREF(5) REFERENCE             CHGIII
C                                                                       OUTPUT
   31 IF(KREF(J,5).EQ.0) KRF = NVEH                                     TTHKREF
      IF(KREF(J,5).NE.0) KRF = KREF(J,5)                                TTHKREF
      CALL DOT31 (D(1,1,L),WMEG(1,L),T1)                                CHGIII
      CALL MAT31 (D(1,1,KRF),T1,T2)                                     CHGIII
      DO 32 I=1,3                                                       OUTPUT
      IF (KRF.NE.L) T2(I)=T2(I)-WMEG(I,KRF)                             PLTINC
   32 T3(I) = T2(I)/(2.0*PI)                                            PLTINC
      IF(LPMI(KRF).EQ.0) GO TO 449                                      CHGIII
      CALL DOT31(DPMI(1,1,KRF),T3,ACC(1,J))                             CHGIII
      GO TO 483                                                         CHGIII
  449 CONTINUE                                                          CHGIII
      DO 457 KJL=1,3                                                    CHGIII
  457 ACC(KJL,J) = T3(KJL)                                              CHGIII
  483 ACC(4,J) = DSQRT(ACC(1,J)**2+ACC(2,J)**2+ACC(3,J)**2)             CHGIII
      GO TO 38                                                          OUTPUT
C                                                                       OUTPUT
C     6. SEGMENT REL. ANGULAR DISPLACEMENT IN KREF(6) REFERENCE         CHGIII
C                                                                       OUTPUT
   34 IF(KREF(J,6).EQ.0) KRF = NVEH                                     TTHKREF
      IF(KREF(J,6).NE.0) KRF = KREF(J,6)                                TTHKREF
      IF (LPMI(KRF).EQ.0.AND.LPMI(L).EQ.0) GO TO 36                     CHGIII
      IF (LPMI(L).EQ.0) GO TO 435                                       CHGIII
      CALL DOT33(DPMI(1,1,L),D(1,1,L),T4)                               CHGIII
```

```
  435 IF (LPMI(KRF).EQ.0) GO TO 436                                   CHGIII
      CALL DOT33(DPMI(1,1,KRF),D(1,1,KRF),T5)                         CHGIII
  436 IF (LPMI(L).NE.0) GO TO 438                                     CHGIII
      CALL DOTT33(D(1,1,L),T5,T1)                                     CHGIII
      GO TO 37                                                        CHGIII
  438 IF (LPMI(KRF).NE.0) GO TO 439                                   CHGIII
      CALL DOTT33(T4,D(1,1,KRF),T1)                                   CHGIII
      GO TO 37                                                        CHGIII
  439 CALL DOTT33(T4,T5,T1)                                           CHGIII
      GO TO 37                                                        CHGIII
   36 CALL DOTT33(D(1,1,L),D(1,1,KRF),T1)                             CHGIII
   37 CALL YPRDEG(T1,ACC(1,J))                                        OUTPUT
      TRACE = 0.5*(T1(1)+T2(2)+T3(3)-1.0)                             OUTPUT
      IF (TRACE.GT. 1.0) TRACE =  1.0                                 OUTPUT
      IF (TRACE.LT.-1.0) TRACE = -1.0                                 OUTPUT
      ACC(4,J) = DACOS(TRACE)/RADIAN                                  OUTPUT
      GO TO 38                                                        OUTPUT
C                                                                     OUTPUT
C     7. JOINT PARAMETERS                                             OUTPUT
C                                                                     OUTPUT
   35 ACC(1,J) = PRJNT(1,L)                                           OUTPUT
      ACC(2,J) = PRJNT(2,L)/RADIAN                                    OUTPUT
      ACC(3,J) = PRJNT(3,L)/RADIAN                                    OUTPUT
      ACC(4,J) = PRJNT(4,L)/RADIAN                                    OUTPUT
      ACC(5,J) = DSQRT(PRJNT(5,L))                                    OUTPUT
      ACC(6,J) = DSQRT(PRJNT(6,L))                                    OUTPUT
      ACC(7,J) = DSQRT(PRJNT(7,L))                                    OUTPUT
      GOTO 38                                                         WINDOP
C                                                                     WINDOP
C     8. SEGMENT WIND FORCE IN KREF(8) REFERENCE                      WINDOP
C                                                                     WINDOP
  601 IF(KREF(J,8).EQ.0) KRF = NGRND                                  TTHKREF
      IF(KREF(J,8).NE.0) KRF = KREF(J,8)                              TTHKREF
      CALL MAT31 (D(1,1,KRF),WF(1,L),T2)                              WINDOP
      IF(LPMI(KRF).EQ.0) GO TO 602                                    WINDOP
      CALL DOT31(DPMI(1,1,KRF),T2,ACC(1,J))                           WINDOP
      GO TO 604                                                       WINDOP
  602 CONTINUE                                                        WINDOP
      DO 603 KJL=1,3                                                  WINDOP
  603 ACC(KJL,J) = T2(KJL)                                            WINDOP
  604 ACC(4,J) = DSQRT(ACC(1,J)**2+ACC(2,J)**2+ACC(3,J)**2)           WINDOP
   38 CONTINUE                                                        OUTPUT
      IF  (.NOT.LTAPE8)  GO TO 40                                     OUTPUT
      KK = 0                                                          OUTPUT
      I2 = 4                                                          OUTPUT
      IF  (K.EQ.7)  I2 = 7                                            OUTPUT
      DO  39  J=J1,J2                                                 OUTPUT
      DO  39  I=1,I2                                                  OUTPUT
      KK = KK+1                                                       OUTPUT
   39 TDATA(KK,NT-20) = ACC(I,J)                                      OUTPUT
```

```
   40 IF  (.NOT.LTHIST)  GO TO 43                                     OUTPUT
      IF (K.LE.6) WRITE (NT,41) USEC,((ACC(I,J),I=1,4),J=J1,J2)       OUTPUT
      IF (K.EQ.8) WRITE (NT,41) USEC,((ACC(I,J),I=1,4),J=J1,J2)       WINDOP
   41 FORMAT(F9.3,3(3X,4F9.3) )                                       OUTPUT
      IF (K.EQ.7) WRITE (NT,42) USEC,((ACC(I,J),I=1,7),J=J1,J2)       OUTPUT
   42 FORMAT(F9.3,2(F5.0,3F9.3,2X,3F9.3))                             OUTPUT
   43 CONTINUE                                                        OUTPUT
      GO TO 44                                                        CHGIII
C                                                                     ATBIII
C     9. JOINT FORCES & TORQUES IN KREF(9) GEOMETRIC COORDINATE SYSTEM WINDOP
C                                                                     CHGIII
  440 DO 860 L=1,KSG                                                  PLTINC
      KRF = NVEH                                                      PLTINC
      IF(KREF(L,9).NE.0) KRF = KREF(L,9)                              PLTINC
      LL=MSG(L,K)                                                     CHGIII
      IF (LPMI(KRF).EQ.0) GO TO 851                                   CHGIII
      CALL DOT33 (DPMI(1,1,KRF),D(1,1,KRF),T5)                        CHGIII
      CALL MAT31 (T5,F(1,LL),T1)                                      CHGIII
      CALL MAT31 (T5,TQ(1,LL),T2)                                     CHGIII
      DO 852 JJ=1,3                                                   CHGIII
      T1(JJ) = T1(JJ)/100.0                                           CHGIII
  852 T2(JJ) = -T2(JJ)/100.0                                          OUT385
      GO TO 859                                                       CHGIII
  851 CONTINUE                                                        CHGIII
      CALL MAT31 (D(1,1,KRF),F(1,LL),T1)                              CHGIII
      CALL MAT31 (D(1,1,KRF),TQ(1,LL),T2)                             CHGIII
      DO 853 JJ=1,3                                                   CHGIII
      T1(JJ) = T1(JJ)/100.0                                           CHGIII
  853 T2(JJ) = -T2(JJ)/100.0                                          OUT385
  859 NT = NT + 1                                                     CHGIII
C     P & E CARRIAGE CONTROL                                          PECONV
      CALL CARCON(NT,1)                                               PECONV
      IF (.NOT.LTAPE8) GO TO 855                                      CHGIII
      DO 854  JL=1,3                                                  CHGIII
      TDATA (JL  ,NT-20) = T1(JL)                                     CHGIII
  854 TDATA (JL+3,NT-20) = T2(JL)                                     CHGIII
  855 CONTINUE                                                        CHGIII
      IF (LTHIST) WRITE (NT,857) USEC,T1,T2                           CHGIII
  857 FORMAT(F9.3,3X,3F9.3,3X,3(2X,D10.3))                            CHGIII
  860 CONTINUE                                                        CHGIII
   44 CONTINUE                                                        CHGIII
C                                                                     ATBIII
C     10.  PRINT BODY PROPERTIES                                      WINDOP
C                                                                     ATBIII
      IF (MCG.EQ.0) GO TO 131                                         ATBIII
      DO 130 NCG=1,MCG                                                ATBIII
      M = MCGIN(1,NCG)                                                ATBIII
      N = MCGIN(2,NCG)                                                ATBIII
      DO 120 J=1,9                                                    ATBIII
  120 T4(J) = 0.0                                                     ATBIII
```

```
      SUMW = 0.0                                                        ATBIII
      T7(1)=0.0                                                         KINETIC
      T7(2)=0.0                                                         KINETIC
      DO 123 I=1,N                                                      ATBIII
      K = MCGIN(I+2,NCG)                                                ATBIII
      WG = W(K)/G                                                       ATBIII
      V=(SEGLV(1,K)-SEGLV(1,M))**2                                      KINETIC
     * +(SEGLV(2,K)-SEGLV(2,M))**2                                      KINETIC
     * +(SEGLV(3,K)-SEGLV(3,M))**2                                      KINETIC
      T7(1)=T7(1)+0.5*WG*V                                              KINETIC
      SUMW = SUMW + WG                                                  ATBIII
      DO 121 J=1,3                                                      ATBIII
      T7(2)=T7(2)+0.5*PHI(J,K)*(WMEG(J,K)-WMEG(J,M))**2                 KINETIC
  121 T1(J) = PHI(J,K)*WMEG(J,K)                                        ATBIII
      CALL DOT31 (D(1,1,K),T1,T2)                                       ATBIII
      CALL CROSS (SEGLP(1,K),SEGLV(1,K),T1)                             ATBIII
      DO 122 J=1,3                                                      ATBIII
      T4(J  ) = T4(J  ) + WG*SEGLP(J,K)                                 ATBIII
      T4(J+3) = T4(J+3) + WG*SEGLV(J,K)                                 ATBIII
  122 T4(J+6) = T4(J+6) + WG*T1(J) + T2(J)                              ATBIII
  123 CONTINUE                                                          ATBIII
      T7(3)=T7(1)+T7(2)                                                 KINETIC
      DO 124 J=1,3                                                      ATBIII
  124 T4(J) = T4(J)/SUMW - SEGLP(J,M)                                   ATBIII
C                                                                       ATBIII
C     TRANSFORM FROM PRINCIPAL AXES TO LOCAL AXES                       TGMOD1
C                                                                       ATBIII
      IF (LPMI(M).EQ.0) GO TO 330                                       ATBIII
      CALL DOT33(DPMI(1,1,M),D(1,1,M),T5)                               ATBIII
      CALL MAT31(T5,T4(1),T1)                                           ATBIII
      CALL MAT31(T5,T4(4),T2)                                           ATBIII
      CALL MAT31(T5,T4(7),T3)                                           ATBIII
      GO TO 333                                                         ATBIII
  330 CONTINUE                                                          ATBIII
      CALL MAT31 (D(1,1,M),T4(1),T1)                                    ATBIII
      CALL MAT31 (D(1,1,M),T4(4),T2)                                    ATBIII
      CALL MAT31 (D(1,1,M),T4(7),T3)                                    ATBIII
  333 CONTINUE                                                          ATBIII
      NT = NT + 1                                                       ATBIII
      IF (.NOT.LTAPE8) GO TO 126                                        ATBIII
      DO 125 J=1,3                                                      ATBIII
      TDATA (J  ,NT-20) = T1(J)                                         ATBIII
      TDATA (J+3,NT-20) = T2(J)                                         ATBIII
      TDATA(J+9,NT-20) = T7(J)                                          KINETIC
  125 TDATA(J+6,NT-20) = T3(J)                                          ATBIII
  126 IF (LTHIST) WRITE (NT,127) USEC,T1,T2,T3,T7                       KINETIC
  127 FORMAT (F9.3,3F8.3,9(1X,D10.3))                                   KINETIC
  130 CONTINUE                                                          ATBIII
  131 CONTINUE                                                          ATBIII
C                                                                       OUTPUT
```

```
C     PRINT PLANE FORCES                                                OUTPUT
C                                                                       OUTPUT
      MPSF = 0                                                          OUTPUT
      IF  (NPL.EQ.0)  GO TO 49                                          OUTPUT
      IF (NPRT(18).EQ.1.OR.NPRT(18).EQ.7) GO TO 49                      VARTTH
      IF (NPRT(18).EQ.10.OR.NPRT(18).EQ.11) GO TO 49                    VARTTH
      IF (NPRT(18).GE.14) GO TO 49                                      VARTTH
      DO  45  J=1,NPL                                                   OUTPUT
   45 MPSF = MPSF + MNPL(J)                                             OUTPUT
      IF  (MPSF.EQ.0)  GO TO 49                                         OUTPUT
      DO 47 J1=1,MPSF,2                                                 OUTPUT
      J2 = MIN0(J1+1,MPSF)                                              OUTPUT
      NT = NT+1                                                         OUTPUT
C     SETUP LOGICAL UNIT CONTROL (PRINTER CONTROL) FOR P & E            PECONV
      CALL CARCON(NT,1)                                                 PECONV
      IF  (.NOT.LTAPE8)  GO TO 47                                       OUTPUT
      KK = 0                                                            OUTPUT
      DO  46  J=J1,J2                                                   OUTPUT
      DO  46  I=1,7                                                     OUTPUT
      KK = KK+1                                                         OUTPUT
   46 TDATA(KK,NT-20) = PSF(I,J)                                        OUTPUT
   47 IF  (LTHIST)  WRITE (NT,48) USEC,((PSF(I,J),I=1,7),J=J1,J2)       OUTPUT
   48 FORMAT(F9.3,2(F9.3,3F9.2,3F8.3) )                                 OUTPUT
C                                                                       OUTPUT
C     PRINT BELT FORCES                                                 OUTPUT
C                                                                       OUTPUT
   49 MBSF = 0                                                          OUTPUT
      IF  (NBLT.EQ.0)  GO TO 67                                         OUTPUT
      IF (NPRT(18).EQ.2.OR.NPRT(18).GE.13) GO TO 67                     VARTTH
      IF (NPRT(18).GE.7.AND.NPRT(18).LE.9) GO TO 67                     VARTTH
      DO  50  J=1,NBLT                                                  OUTPUT
   50 MBSF = MBSF + MNBLT(J)                                            OUTPUT
      IF  (MBSF.EQ.0)  GO TO 67                                         OUTPUT
      DO 52 J1=1,MBSF,2                                                 OUTPUT
      J2 = MIN0(J1+1,MBSF)                                              OUTPUT
      NT = NT+1                                                         OUTPUT
C     LOGICAL UNIT (PRINTER CONTROL) FOR P & E                          PECONV
      CALL CARCON(NT,1)                                                 PECONV
      IF  (.NOT.LTAPE8)  GO TO 52                                       OUTPUT
      KK = 0                                                            OUTPUT
      DO  51  J=J1,J2                                                   OUTPUT
      DO  51  I=1,4                                                     OUTPUT
      KK = KK+1                                                         OUTPUT
   51 TDATA(KK,NT-20) = BSF(I,J)                                        OUTPUT
   52 IF  (LTHIST)  WRITE (NT,53) USEC,((BSF(I,J),I=1,4),J=J1,J2)       OUTPUT
   53 FORMAT(F9.3,4(F15.6,F12.2,3X) )                                   OUTPUT
C                                                                       OUTPUT
C     PRINT HARNESS-BELT ENDPOINT FORCES (STORED IN BSF ARRAY).         OUTPUT
C                                                                       OUTPUT
   67 IF  (NHRNSS.LE.0)  GO TO 71                                       OUTPUT
```

```
      IF (NPRT(18).EQ.3.OR.NPRT(18).EQ.11) GO TO 71                     VARTTH
      IF (NPRT(18).EQ.9.OR.NPRT(18).EQ.8) GO TO 71                      VARTTH
      IF (NPRT(18).EQ.13.OR.NPRT(18).EQ.14) GO TO 71                    VARTTH
      IF (NPRT(18).GE.16) GO TO 71                                      VARTTH
      MBSF1 = MBSF + 1                                                  OUTPUT
      DO  68  I=1,NHRNSS                                                OUTPUT
   68 MBSF = MBSF + NBLTPH(I)                                           OUTPUT
      DO  70  J1=MBSF1,MBSF,2                                           OUTPUT
      J2 = MIN0(J1+1,MBSF)                                              OUTPUT
      NT = NT+1                                                         OUTPUT
C     LOGICAL UNIT (PRINTER CONTROL) FOR P & E                          PECONV
      CALL CARCON(NT,1)                                                 PECONV
      IF  (.NOT.LTAPE8)  GO TO 70                                       OUTPUT
      KK = 0                                                            OUTPUT
      DO  69  J=J1,J2                                                   OUTPUT
      DO  69  I=1,4                                                     OUTPUT
      KK = KK+1                                                         OUTPUT
   69 TDATA(KK,NT-20) = BSF(I,J)                                        OUTPUT
   70 IF  (LTHIST)  WRITE (NT,53) USEC,((BSF(I,J),I=1,4),J=J1,J2)       OUTPUT
C                                                                       OUTPUT
C     PRINT SPRING DAMPER FORCES (STORED IN BSF ARRAY).                 OUTPUT
C                                                                       OUTPUT
   71 IF  (NSD.LE.0)  GO TO 54                                          OUTPUT
      IF (NPRT(18).EQ.4.OR.NPRT(18).EQ.9) GO TO 54                      VARTTH
      IF (NPRT(18).GE.12) GO TO 54                                      VARTTH
      MBSF1 = MBSF + 1                                                  OUTPUT
      MBSF = MBSF + (NSD+1)/2                                           OUTPUT
      DO  73  J1=MBSF1,MBSF,2                                           OUTPUT
      J2 = MIN0(J1+1,MBSF)                                              OUTPUT
      NT = NT+1                                                         OUTPUT
C     LOGICAL UNIT (PRINTER CONTROL) FOR P & E                          PECONV
      CALL CARCON(NT,1)                                                 PECONV
      IF  (.NOT.LTAPE8)  GO TO 73                                       OUTPUT
      KK = 0                                                            OUTPUT
      DO  72  J=J1,J2                                                   OUTPUT
      DO  72  I=1,4                                                     OUTPUT
      KK = KK+1                                                         OUTPUT
   72 TDATA(KK,NT-20) = BSF(I,J)                                        OUTPUT
   73 IF  (LTHIST)  WRITE (NT,74) USEC,((BSF(I,J),I=1,4),J=J1,J2)       OUTPUT
   74 FORMAT (F9.3,4(F14.3,F12.2,4X))                                   OUTPUT
C                                                                       OUTPUT
C     PRINT SEGMENT CONTACT FORCES                                      OUTPUT
C                                                                       OUTPUT
   54 MSSF = 0                                                          OUTPUT
      IF (NPRT(18).EQ.5.OR.NPRT(18).EQ.13) GO TO 161                    VARTTH
      IF (NPRT(18).EQ.10.OR.NPRT(18).EQ.11) GO TO 161                   VARTTH
      IF (NPRT(18).GE.15) GO TO 161                                     VARTTH
      DO  55  J=1,NSEG                                                  OUTPUT
   55 MSSF = MSSF + MNSEG(J)                                            OUTPUT
      IF  (MSSF.EQ.0)  GO TO 59                                         OUTPUT
```

```
      DO 57 J=1,MSSF                                                    OUTPUT
      NT = NT+1                                                         OUTPUT
C     LOGICAL UNIT (PRINTER CONTROL) FOR P & E                          PECONV
      CALL CARCON(NT,1)                                                 PECONV
      IF  (.NOT.LTAPE8)  GO TO 57                                       OUTPUT
      DO  56  I=1,10                                                    OUTPUT
   56 TDATA(I,NT-20) = SSF(I,J)                                         OUTPUT
   57 IF  (LTHIST)  WRITE (NT,58) USEC,(SSF(I,J),I=1,10)                OUTPUT
   58 FORMAT(2F9.3,3F9.2,3F8.3,2X,3F8.3)                                OUTPUT
  161 CONTINUE                                                          VARTTH
C                                                                       OUTPUT
C     PRINT AIRBAG FORCES                                               OUTPUT
C                                                                       OUTPUT
   59 IF (NBAG.EQ.0) GO TO 65                                           OUTPUT
      IF (NPRT(18).EQ.6.OR.NPRT(18).EQ.9) GO TO 65                      VARTTH
      IF (NPRT(18).GE.12) GO TO 65                                      VARTTH
      K1 = 1                                                            OUTPUT
      DO 64 J=1,NBAG                                                    OUTPUT
      IF (MNBAG(J).EQ.0) GO TO 64                                       OUTPUT
      KBAG = MNBAG(J)+NPANEL(J)+5                                       OUTPUT
      DO 63 J1=1,KBAG,4                                                 OUTPUT
      J2 = MIN0(J1+3,KBAG)                                              OUTPUT
      K2 = K1+J2-J1                                                     OUTPUT
      NT = NT+1                                                         OUTPUT
C     LOGICAL UNIT (PRINTER CONTROL) FOR P & E                          PECONV
      CALL CARCON(NT,1)                                                 PECONV
      IF  (.NOT.LTAPE8)  GO TO 61                                       OUTPUT
      KK = 0                                                            OUTPUT
      DO  60  K=K1,K2                                                   OUTPUT
      DO  60  I=1,3                                                     OUTPUT
      KK = KK+1                                                         OUTPUT
   60 TDATA(KK,NT-20) = BAGSF(I,K)                                      OUTPUT
   61 IF  (.NOT.LTHIST)  GO TO 63                                       OUTPUT
      IF  (J1.EQ.1)  WRITE (NT,75) USEC,((BAGSF(I,K),I=1,3),K=K1,K2)    OUTPUT
      IF  (J1.NE.1)  WRITE (NT,62) USEC,((BAGSF(I,K),I=1,3),K=K1,K2)    OUTPUT
   75 FORMAT (F9.3,3X,3F9.2,2(3X,3F9.3),3X,3F9.2)                       OUTPUT
   62 FORMAT(F9.3,4(3X,3F9.2))                                          OUTPUT
   63 K1 = K2+1                                                         OUTPUT
   64 CONTINUE                                                          OUTPUT
   65 NT = NT-20                                                        OUTPUT
      IF(NT.GT.NTMAX) STOP 56                                           CHGIII
      IF  (LTAPE8)  WRITE (8) NT,USEC,((TDATA(I,J),I=1,14),J=1,NT)      OUTPUT
      PREVT = TIME                                                      OUTPUT
      CALL ELTIME(2,8)                                                  OUTPUT
   66 RETURN                                                            OUTPUT
      END                                                               OUTPUT
```

```
      SUBROUTINE PANEL (DRR,ZR,JB)                                      PANEL
C                                                    REV III.2 08/08/84REVIII
C     COMPUTES AIRBAG PARAMETERS DURING INFLATION OF BAG.               PANEL
C                                                                       PANEL
C         GIVEN: DRR - DC MATRIX RELATIVE TO VEHICLE                    PANEL
C                 ZR - CG LOCATION IN VEHICLE REFERENCE                 PANEL
C                                                                       PANEL
C       COMPUTE: SEGLP,SEGLV,SEGLA,D,WMEG & WMEGD FOR SEGMENT JB.       PANEL
C                                                                       PANEL
      IMPLICIT REAL*8 (A-H,O-Z)                                         PANEL
      DIMENSION DRR(3,3),ZR(3),T1(3),T2(3)                              PANEL
      COMMON/CONTRL/ TIME,NSEG,NJNT,NPL,NBLT,NBAG,NVEH,NGRND,           PANEL
     *               NS,NQ,NSD,NFLX,NHRNSS,NWINDF,NJNTF,NPRT(36),NPG    PAGE
      COMMON/SGMNTS/ D(3,3,30),WMEG(3,30),WMEGD(3,30),U1(3,30),U2(3,30),PANEL
     *               SEGLP(3,30),SEGLV(3,30),SEGLA(3,30),NSYM(30)       PANEL
      CALL MAT33 (DRR,D(1,1,NVEH),D(1,1,JB))                            PANEL
      CALL MAT31 (DRR,WMEG(1,NVEH),WMEG(1,JB))                          PANEL
      CALL DOT31 (D(1,1,NVEH),ZR,SEGLP(1,JB))                           PANEL
      CALL CROSS (WMEG(1,NVEH),ZR,T1)                                   PANEL
      CALL DOT31 (D(1,1,NVEH),T1,SEGLV(1,JB))                           PANEL
      CALL CROSS (WMEG(1,NVEH),T1,T2)                                   PANEL
      CALL DOT31 (D(1,1,NVEH),T2,SEGLA(1,JB))                           PANEL
      DO  10  I=1,3                                                     PANEL
      SEGLP(I,JB) = SEGLP(I,JB) + SEGLP(I,NVEH)                         PANEL
      SEGLV(I,JB) = SEGLV(I,JB) + SEGLV(I,NVEH)                         PANEL
      SEGLA(I,JB) = SEGLA(I,JB) + SEGLA(I,NVEH)                         PANEL
   10 WMEGD(I,JB) = WMEGD(I,NVEH)                                       PANEL
      RETURN                                                            PANEL
      END                                                               PANEL
```

```
      SUBROUTINE PDAUX (VAR,DER,NEQ,KDINT)                              PDAUX
C                                                    REV IV    07/24/86SLIP
C     PURPOSE IS TO ACT AS INTERFACE BETWEEN INTEGRATOR AND DAUX TO     PDAUX
C     ACCOMODATE VARIABLE NUMBER OF FUNCTIONS TO BE INTEGRATED.         PDAUX
C                                                                       PDAUX
C     ARGUMENTS:                                                        PDAUX
C        VAR - ARRAY OF NEQ STATE VARIABLES UPDATED BY DINT.            PDAUX
C        DER - ARRAY OF NEQ DERIVATIVES TO BE SUPPLIED BY DAUX.         PDAUX
C        NEQ - NUMBER OF STATE VARIABLES AND DERIVATIVES.               PDAUX
C      KDINT - INTEGRATION STEP NUMBER IN DINT.                         PDAUX
C                                                                       PDAUX
      IMPLICIT REAL*8 (A-H,O-Z)                                         PDAUX
      DIMENSION VAR(3,1),DER(3,1)                                       PDAUX
      COMMON/CONTRL/ TIME,NSEG,NJNT,NPL,NBLT,NBAG,NVEH,NGRND,           ?DAUX
     *               NS,NQ,NSD,NFLX,NHRNSS,NWINDF,NJNTF,NPRT(36),NPG    PAGE
      COMMON/SGMNTS/ D(3,3,30),WMEG(3,30),WMEGD(3,30),U1(3,30),U2(3,30),PDAUX
     *               SEGLP(3,30),SEGLV(3,30),SEGLA(3,30),NSYM(30)       PDAUX
      COMMON/DESCRP/ PHI(3,30),W(30),RW(30),SR(4,60),HA(3,60),HB(3,60), SLIP
     *               RPHI(3,30),HT(3,3,60),SPRING(5,90),VISC(7,90),     PDAUX
     *               JNT(30),IPIN(30),ISING(30),IGLOB(30),JOINTF(30)    PDAUX
      COMMON/TITLES/ DATE(3),COMENT(40),VPSTTL(20),BDYTTL(5),           PDAUX
     *               BLTTTL(5,8),PLTTL(5,30),BAGTTL(5,6),SEG(30),       PDAUX
     *               JOINT(30),CGS(30),JS(30)                           PDAUX
      REAL DATE,COMENT,VPSTTL,BDYTTL,BLTTTL,PLTTL,BAGTTL,SEG,JOINT      PDAUX
      LOGICAL*1 CGS,JS                                                  PDAUX
      COMMON/INTEST/ SGTEST(3,4,30),XTEST(3,120),SEGT(120),REGT(120)    PDAUX
      REAL    SEGT                                                      PDAUX
      COMMON/FLXBLE/ HF(4,12,8),B42(3,3,24),V4(3,8),NFLEX(3,8)          PDAUX
      COMMON/CEULER/ IEULER(30),HIR(3,3,90),ANG(3,30),ANGD(3,30),       SLIP
     *               FE(3,30),TQE(3,30),CONST(5,30)                     SLIP
      COMMON/TEMPVS/ T(3,30),VXT(3)                                     PDAUX
      DIMENSION SD(3,3,30) , E1(30) , NTST(30) , LSEG(30) , RGTTL(4)    PDAUX
      LOGICAL  LSEG                                                     PDAUX
      DATA NTST/30*0/                                                   PDAUX
      DATA RGTTL/8HANG VEL ,8HLIN VEL ,8HANG ACC ,8HLIN ACC /           PDAUX
      CALL ELTIME(1,6)                                                  PDAUX
      MBAG = NGRND                                                      PDAUX
      IF (NTST(1).NE.0) GO TO 10                                        PDAUX
      LSEG(1) = .FALSE.                                                 VAXCHG
      NTST(1) = 1                                                       ATBIII
      DO 5 M=2,MBAG                                                     ATBIII
      LSEG(M) = ISING(M).GE.0 .AND. JNT(M-1).NE.0                       ATBIII
      IF (IABS(IPIN(M-1)).GE.5.AND.IEULER(M-1).GE.0) LSEG(M)=.FALSE.    SLIP
    5 NTST(M) = M                                                       PDAUX
      NTST(NGRND) = -NGRND                                              PDAUX
      LSEG(NGRND) = .TRUE.                                              PDAUX
      IF (NFLX.EQ.0) GO TO 10                                           PDAUX
      DO 6 J=1,NFLX                                                     PDAUX
      M = NFLEX(2,J)                                                    PDAUX
    6 NTST(M) = -M                                                      PDAUX
```

```
   10 IF  (KDINT.EQ.4)  GO TO 48                                    PDAUX
      IF  (KDINT.GT.0)  GO TO 20                                    PDAUX
C                                                                   PDAUX
C     KDINT=0 IMPLIES INITIAL CALL FROM DINT. PDAUX TO SUPPLY INITIAL PDAUX
C     VALUES TO STATE VARIABLES AND COMPUTE VALUE OF NEQ.           PDAUX
C                                                                   PDAUX
C                                                                   PDAUX
C        (A) SET Q TO IDENTITY QUATERNION                           PDAUX
C                                                                   PDAUX
      N = 0                                                         PDAUX
      DO 12 M=1,MBAG                                                PDAUX
      IF (NTST(M).LT.0) GO TO 12                                    PDAUX
      N = N+1                                                       PDAUX
      REGT(N) = RGTTL(1)                                            PDAUX
      SEGT(N) = SEG(M)                                              PDAUX
      E1(N) = 1.0                                                   PDAUX
      DO 11 I=1,3                                                   PDAUX
      XTEST(I,N) = SGTEST(I,1,M)**2                                 PDAUX
   11 VAR(I,N) = 0.0                                                PDAUX
   12 CONTINUE                                                      PDAUX
C                                                                   PDAUX
C        (B) SEGLP OF REFERENCE SEGMENTS                            PDAUX
C                                                                   PDAUX
      DO 14 M=1,MBAG                                                PDAUX
      IF  (LSEG(M))  GO TO 14                                       PDAUX
      N = N+1                                                       PDAUX
      REGT(N) = RGTTL(2)                                            PDAUX
      SEGT(N) = SEG(M)                                              PDAUX
      DO 13 I=1,3                                                   PDAUX
      XTEST(I,N) = SGTEST(I,2,M)**2                                 PDAUX
   13 VAR(I,N) = SEGLP(I,M)                                         PDAUX
   14 CONTINUE                                                      PDAUX
C                                                                   PDAUX
C        (C) WMEG                                                   PDAUX
C                                                                   PDAUX
      DO 16 M=1,MBAG                                                PDAUX
      IF (NTST(M).LT.0) GO TO 16                                    PDAUX
      N = N+1                                                       PDAUX
      REGT(N) = RGTTL(3)                                            PDAUX
      SEGT(N) = SEG(M)                                              PDAUX
      DO 15 I=1,3                                                   PDAUX
      XTEST(I,N) = SGTEST(I,3,M)**2                                 PDAUX
   15 VAR(I,N) = WMEG(I,M)                                          PDAUX
   16 CONTINUE                                                      PDAUX
C                                                                   PDAUX
C        (D) SEGLV OF REFERENCE SEGMENTS                            PDAUX
C                                                                   PDAUX
      DO 18 M=1,MBAG                                                PDAUX
      IF  (LSEG(M))  GO TO 18                                       PDAUX
      N = N+1                                                       PDAUX
```

```
      REGT(N) = RGTTL(4)                                                PDAUX
      SEGT(N) = SEG(M)                                                  PDAUX
      DO 17 I=1,3                                                       PDAUX
      XTEST(I,N) = SGTEST(I,4,M)**2                                     PDAUX
   17 VAR(I,N) = SEGLV(I,M)                                             PDAUX
   18 CONTINUE                                                          PDAUX
      NEQ = 3*N                                                         PDAUX
      GO TO 40                                                          PDAUX
   20 IF (KDINT.NE.1) GO TO 30                                          PDAUX
C                                                                       PDAUX
C     KDINT = 1, 1ST STEP IN ADVANCING INTEGRATING INTERVAL,            PDAUX
C                    SAVE DC MATRICES IF TIME HAS ADVANCED.             PDAUX
C                                                                       PDAUX
      N = 0                                                             PDAUX
      DO 22 M=1,MBAG                                                    PDAUX
      IF (NTST(M).LT.0) GO TO 22                                        PDAUX
      N = N+1                                                           PDAUX
      DO 21 J=1,3                                                       PDAUX
      DO 21 I=1,3                                                       PDAUX
   21 SD(I,J,N) = D(I,J,M)                                              PDAUX
   22 CONTINUE                                                          PDAUX
C                                                                       PDAUX
C     KDINT > 0,1 - FETCH SAVED DC MATRICES AND UPDATE BY CURRENT THETA.PDAUX
C                                                                       PDAUX
C       (A) UPDATE D BY Q                                               PDAUX
C                                                                       PDAUX
   30 N = 0                                                             PDAUX
      DO 32 M=1,MBAG                                                    PDAUX
      IF (NTST(M).LT.0) GO TO 32                                        PDAUX
      N = N+1                                                           PDAUX
      EDOTE = VAR(1,N)**2 + VAR(2,N)**2 + VAR(3,N)**2                   PDAUX
      IF (EDOTE.GE.1.0) KDINT = -KDINT                                  PDAUX
      IF (KDINT.LE.0)  GO TO 99                                         PDAUX
      E1(N) = DSQRT(1.0-EDOTE)                                          PDAUX
      CALL DSETQ(SD(1,1,N),VAR(1,N),EDOTE,E1(N),D(1,1,M))               PDAUX
   32 CONTINUE                                                          PDAUX
C                                                                       PDAUX
C     KDINT > 0 - STORE STATE VARIABLES INTO PROGRAM ARRAYS.            PDAUX
C                                                                       PDAUX
C       (B) SEGLP OF REFERENCE SEGMENTS                                 PDAUX
C                                                                       PDAUX
      DO 35 M=1,MBAG                                                    PDAUX
      IF  (LSEG(M))  GO TO 35                                           PDAUX
      N = N+1                                                           PDAUX
      DO 34 I=1,3                                                       PDAUX
   34 SEGLP(I,M) = VAR(I,N)                                             PDAUX
   35 CONTINUE                                                          PDAUX
C                                                                       PDAUX
C       (C) WMEG                                                        PDAUX
C                                                                       PDAUX
```

```
      DO 31 M=1,MBAG                                                    PDAUX
      IF (NTST(M).LT.0) GO TO 31                                        PDAUX
      N = N+1                                                           PDAUX
      DO 36 I=1,3                                                       PDAUX
   36 WMEG(I,M) = VAR(I,N)                                              PDAUX
   31 CONTINUE                                                          PDAUX
C                                                                       PDAUX
C        (D) SEGLV OF REFERENCE SEGMENTS                                PDAUX
C                                                                       PDAUX
      DO 38 M=1,MBAG                                                    PDAUX
      IF  (LSEG(M))  GO TO 38                                           PDAUX
      N = N+1                                                           PDAUX
      DO 37 I=1,3                                                       PDAUX
   37 SEGLV(I,M) = VAR(I,N)                                             PDAUX
   38 CONTINUE                                                          PDAUX
C                                                                       PDAUX
C     CALL DAUX ROUTINE TO COMPUTE DERIVATIVES                          PDAUX
C                                                                       PDAUX
   40 CALL DAUX(0)                                                      PDAUX
C                                                                       PDAUX
C     STORE DERIVATIVES FOR INTEGRATING SUBROUTINE.                     PDAUX
C                                                                       PDAUX
C        (A) DERIVATIVE OF Q                                            PDAUX
C                                                                       PDAUX
      N = 0                                                             PDAUX
      DO 39 M=1,MBAG                                                    PDAUX
      IF (NTST(M).LT.0) GO TO 39                                        PDAUX
      N = N+1                                                           PDAUX
      CALL CROSS(VAR(1,N),WMEG(1,M),VXT)                                PDAUX
      DO 41 I=1,3                                                       PDAUX
   41 DER(I,N) = 0.5*(E1(N)*WMEG(I,M) + VXT(I) )                        PDAUX
   39 CONTINUE                                                          PDAUX
      NQUAT = N                                                         PDAUX
C                                                                       PDAUX
C        (B) SEGLV OF REFERENCE SEGMENTS                                PDAUX
C                                                                       PDAUX
      DO 43 M=1,MBAG                                                    PDAUX
      IF  (LSEG(M))  GO TO 43                                           PDAUX
      N = N+1                                                           PDAUX
      DO 42 I=1,3                                                       PDAUX
   42 DER(I,N) = SEGLV(I,M)                                             PDAUX
   43 CONTINUE                                                          PDAUX
C                                                                       PDAUX
C        (C) WMEGD                                                      PDAUX
C                                                                       PDAUX
      DO 47 M=1,MBAG                                                    PDAUX
      IF (NTST(M).LT.0) GO TO 47                                        PDAUX
      N = N+1                                                           PDAUX
      DO 44 I=1,3                                                       PDAUX
   44 DER(I,N) = WMEGD(I,M)                                             PDAUX
```

```
   47 CONTINUE                                                          PDAUX
C                                                                       PDAUX
C        (D) SEGLA OF REFERENCE SEGMENTS                                PDAUX
C                                                                       PDAUX
      DO 46 M=1,MBAG                                                    PDAUX
      IF  (LSEG(M))  GO TO 46                                           PDAUX
      N = N+1                                                           PDAUX
      DO 45 I=1,3                                                       PDAUX
   45 DER(I,N) = SEGLA(I,M)                                             PDAUX
   46 CONTINUE                                                          PDAUX
      IF (KDINT.NE.4) GO TO 99                                          PDAUX
   48 N = 0                                                             PDAUX
      DO 51 M=1,MBAG                                                    PDAUX
      IF (NTST(M).LT.0) GO TO 51                                        PDAUX
      N = N+1                                                           PDAUX
      E1(N) = 1.0                                                       PDAUX
      DO 50 I=1,3                                                       PDAUX
      DER(I,N) = 0.5*WMEG(I,M)                                          PDAUX
   50 VAR(I,N) = 0.0                                                    PDAUX
   51 CONTINUE                                                          PDAUX
   99 IF  (KDINT.EQ.2)  KDINT = NQUAT                                   PDAUX
      CALL ELTIME(2,6)                                                  PDAUX
      RETURN                                                            PDAUX
      END                                                               PDAUX
```

```
      SUBROUTINE PLEDG(AREAL,BD,PL)                                       PLEDG
C                                                   REV IV    12/11/87HYFIX
      IMPLICIT REAL*8(A-H,O-Z)                                            PLEDG
      LOGICAL AREAL                                                       PLEDG
      DIMENSION BD(24),PL(24)                                             HYFIX
      DIMENSION HAREA(2,2,5),ZC(3,14),X(3),UV(3,2),IV(14)                 HYFIX
C SHARED WITH PLELP-PLSEGF                                                HYFIX
        COMMON/TEMPVS/DMNT(3,3),DHNT(3,3),DUM1(18),TM(3),R(3),RM(3),      HYFIX
     X              DUM2(9),UP(3),VP(3),U(3),V(3),EU(3),EV(3),ET(3),      HYFIX
     X              A(2),B(2),CC(2),DUM4(12),TH(3),XH(3),RMD(3),RND(3),   HYFIX
     X              APT(2,2,2),AC(2,2),BC(2,2),AFP,E(2,2),DELT,AREA,      HYFIX
     X              AB,BB,BT(2),XNC(3),UH(3),P,AMR,FM,T4(3),ALIM(2,2)     HYFIX
      EQUIVALENCE (UV(1,1),U(1))                                          HYFIX
      EQUIVALENCE (ALIM(1,1),BMIN),(ALIM(1,2),AMIN)                       HYFIX
      EQUIVALENCE (ALIM(2,1),BMAX),(ALIM(2,2),AMAX)                       HYFIX
      EQUIVALENCE (AC(1,1),BB1),(AC(1,2),AA1)                             HYFIX
      EQUIVALENCE (AC(2,1),BB2),(AC(2,2),AA2)                             HYFIX
      EQUIVALENCE (BC(1,1),AB1),(BC(1,2),BA1)                             HYFIX
      EQUIVALENCE (BC(2,1),AB2),(BC(2,2),BA2)                             HYFIX
C                                                                         HYFIX
      AREA = 0.0                                                          PLEDG
      AREAL = .FALSE.                                                     PLEDG
CALCULATE CENTER OF ELLIPSE IN PLANE                                      PLEDG
C T4 IS VECTOR FROM CENTER OF ELLIPSOID TO CENTER OF ELLIPSE              PLEDG
      DO 10 I = 1,3                                                       PLEDG
      T4(I) = FM*XH(I)                                                    HYFIX
   10 XNC(I) = XNC(I) + T4(I)                                             PLEDG
C XNC P1 TO CENTER OF ELLIPSE                                             PLEDG
C PUT PLANE VECTORS IN ELLIPSE SYSTEM   TH IS PLANE VECTOR                PLEDG
      IF (BD(1).LT.0.0) CALL MAT33(BD(8),DMNT,DHNT)                       HYPER
      IF (BD(1).LT.0.0) GO TO 20                                          HYPER
      DO 15 I = 1,3                                                       HYPER
      DO 15 J = 1,3                                                       HYPER
   15 DHNT(I,J) = DMNT(I,J)                                               HYPER
   20 CALL MAT31(DHNT,PL( 8),UP)                                          HYPER
      CALL MAT31(DHNT,PL(13),VP)                                          HYPER
      CALL MAT31(DHNT,PL(18), U)                                          HYPER
      CALL MAT31(DHNT,PL(21), V)                                          HYPER
C U IS P2 - P1, V IS P3 - P1, PLANE VECTOR IS TM                          PLEDG
CALCULATE CENTER FROM P1 IN U, V COORDINATES                              PLEDG
      B(1) = (UP(1)*XNC(1) + UP(2)*XNC(2) + UP(3)*XNC(3))/PL(12)          PLEDG
      B(2) = (VP(1)*XNC(1) + VP(2)*XNC(2) + VP(3)*XNC(3))/PL(17)          PLEDG
      AMIN = -B(1)                                                        HYFIX
      AMAX = 1.0 - B(1)                                                   HYFIX
      BMIN = -B(2)                                                        HYFIX
      BMAX = 1.0 - B(2)                                                   HYFIX
C GET ELLIPSE EQUATION                                                    PLEDG
      DO 25 I = 1,2                                                       HYPER
      DO 25 J = I,2                                                       HYPER
   25 E(I,J) = 0.0                                                        HYPER
```

```
      IF (BD(1).GT.0.0) GO TO 35                                        HYPER
C TREAT HYPER AS ELLIPSE FOR FIRST GUESS                                HYPER
      DO 30 I = 1,3                                                     HYPER
      EU(I) = U(I)*BD(I+16)                                             HYPER
   30 EV(I) = V(I)*BD(I+16)                                             HYPER
C GET INTERSECTION OF PLANE WITH BOX                                    HYFIX
      CALL HYBOX(BD(2),TH,T4,MB,ZC,IV)                                  HYPER
      IF (MB.LT.6) GO TO 140                                            HYPER
      GO TO 40                                                          HYPER
   35 CALL MAT31(BD(7),U,EU)                                            HYPER
      CALL MAT31(BD(7),V,EV)                                            HYPER
   40 DO 45 K = 1,3                                                     HYPER
      E(1,1) = E(1,1) + U(K)*EU(K)                                      HYPER
      E(1,2) = E(1,2) + V(K)*EU(K)                                      HYPER
   45 E(2,2) = E(2,2) + V(K)*EV(K)                                      HYPER
      DELT = E(1,1)*E(2,2) - E(1,2)**2                                  PLEDG
C WHAT ABOUT AMR FOR HYPER??   1 - FM**P ?                              HYFIX
      R2D = AMR/DELT                                                    HYFIX
COMPUTE BOUNDS OF ELLIPSOID LOCATION OF MAX AND MIN ALPHA               HYFIX
      AA2 =  DSQRT(E(2,2)*R2D)                                          HYFIX
      AA1 = -AA2                                                        HYFIX
C BA IS VALUE OF BETA AT AT ALPHA MAX                                   HYFIX
      BA1 =  E(1,2)*AA2/E(2,2)                                          HYFIX
      BA2 = -BA1                                                        HYFIX
      IF (BD(1).GE.-2.0) GO TO 50                                       HYPER
      CALL HYBND(MB,ZC,IV,UP,-1.,X)                                     HYPER
      CALL HYLIM(AA1,U,BA1,V,FM,XH,X,BD)                                HYFIX
   50 AMIN = DMAX1(AA1,AMIN)                                            HYFIX
      IF (AMIN.GE.AMAX) GO TO 140                                       HYFIX
      IF (BD(1).GE.-2.0) GO TO 55                                       HYPER
      CALL HYBND(MB,ZC,IV,UP, 1.,X)                                     HYPER
      CALL HYLIM(AA2,U,BA2,V,FM,XH,X,BD)                                HYFIX
   55 AMAX = DMIN1(AA2,AMAX)                                            HYFIX
      IF (AMIN.GE.AMAX) GO TO 140                                       HYPER
COMPUTE BOUNDS OF ELLIPSOID LOCATION OF MAX AND MIN BETA                HYFIX
      BB2 =  DSQRT(E(1,1)*R2D)                                          HYFIX
      BB1 = -BB2                                                        HYFIX
C AB IS VALUE OF ALPHA AT AT BETA MAX                                   HYFIX
      AB1 =  E(1,2)*BB2/E(1,1)                                          HYFIX
      AB2 = -AB1                                                        HYFIX
      IF (BD(1).GE.-2.0) GO TO 60                                       HYPER
      CALL HYBND(MB,ZC,IV,VP,-1.,X)                                     HYPER
      CALL HYLIM(BB1,V,AB1,U,FM,XH,X,BD)                                HYFIX
   60 BMIN = DMAX1(BB1,BMIN)                                            HYFIX
      IF (BMIN.GE.BMAX) GO TO 140                                       HYFIX
      IF (BD(1).GE.-2.0) GO TO 65                                       HYPER
      CALL HYBND(MB,ZC,IV,VP, 1.,X)                                     HYPER
      CALL HYLIM(BB2,V,AB2,U,FM,XH,X,BD)                                HYFIX
   65 BMAX = DMIN1(BB2,BMAX)                                            HYFIX
      IF (BMIN.GE.BMAX) GO TO 140                                       HYPER
```

```
COMPUTE ALPHA'S AT BMIN AND BMAX; BETA'S AT AMIN AND AMAX IF NOT ON      HYFIX
C ELLIPSOID                                                              HYFIX
      IF (BD(1).LT.-2.0) GO TO 80                                        HYPER
      DO 76 L = 1,2                                                      HYFIX
      K = 3 - L                                                          HYFIX
      DO 75 J = 1,2                                                      HYPER
      DIS = 0.0                                                          HYFIX
      AFP = BC(J,L)                                                      HYFIX
      IF (ALIM(J,L).EQ.AC(J,L)) GO TO 74                                 HYFIX
      AFP = ALIM(J,L)/E(L,L)                                             HYFIX
      DISC =  AMR/E(L,L) - DELT*AFP**2                                   HYFIX
      DIS = 0.0                                                          HYFIX
      IF (DISC.GT.0.0) DIS = DSQRT(DISC)                                 HYFIX
      AFP =  -AFP*E(1,2)                                                 HYFIX
   74 APT(1,J,L) = DMAX1 (AFP-DIS,ALIM(1,K))                             HYFIX
      APT(2,J,L) = DMIN1 (AFP+DIS,ALIM(2,K))                             HYFIX
   75 CONTINUE                                                           HYPER
   76 CONTINUE                                                           HYFIX
      GO TO 95                                                           HYPER
   80 DO 90 L = 1,2                                                      HYPER
      K = 3 - L                                                          HYFIX
      DO 89 J = 1,2                                                      HYFIX
      DIS = 0.0                                                          HYFIX
      BT(1) = BC(J,L)                                                    HYFIX
      BT(2) = BC(J,L)                                                    HYFIX
      IF (ALIM(J,L).EQ.AC(J,L)) GO TO 88                                 HYFIX
      M = 2                                                              HYFIX
      IF (ALIM(J,L).LT.0.0) M = 1                                        HYFIX
      CM = BC(M,L)/AC(M,L)                                               HYFIX
      CL = ALIM(J,L)*CM                                                  HYFIX
      DO 82 I = 1,3                                                      HYFIX
   82 RM(I) = T4(I) + ALIM(J,L)*(UV(I,K) + CM*UV(I,L))                   HYFIX
      DO 85 I = 1,2                                                      HYFIX
      CALL HYVAL(BT(I),UV(1,L),RM,BD,I)                                  HYFIX
   85 BT(I) = BT(I) + CL                                                 HYFIX
   88 APT(1,J,L) = DMAX1 (BT(1),ALIM(1,K))                               HYFIX
      APT(2,J,L) = DMIN1 (BT(2),ALIM(2,K))                               HYFIX
   89 CONTINUE                                                           HYFIX
   90 CONTINUE                                                           HYPER
C  SET UP LEGAL BOUNDARIES                                               HYFIX
C  APT          L = 1                       L = 2                        HYFIX
C        A-(BMIN)   A-(BMAX)     B-(AMIN)   B-(AMAX)                     HYFIX
C        A+(BMIN)   A+(BMAX)     B+(AMIN)   B+(AMAX)                     HYFIX
C SET UP HAREA (LINE SEGMENTS) CLOCKWISE STARTING WITH AMIN              HYFIX
   95 L = 0                                                              HYFIX
      HAREA(1,1,L+1) = AMIN                                              HYFIX
      HAREA(2,1,L+1) = APT(2,1,2)                                        HYFIX
      HAREA(1,2,L+1) = AMIN                                              HYFIX
      HAREA(2,2,L+1) = APT(1,1,2)                                        HYFIX
      IF (APT(2,1,2).GE.APT(1,1,2)) L = L + 1                            HYFIX
```

```
      HAREA(1,1,L+1) = APT(1,1,1)                                          HYFIX
      HAREA(2,1,L+1) = BMIN                                                HYFIX
      HAREA(1,2,L+1) = APT(2,1,1)                                          HYFIX
      HAREA(2,2,L+1) = BMIN                                                HYFIX
      IF (APT(2,1,1).GE.APT(1,1,1)) L = L + 1                              HYFIX
      HAREA(1,1,L+1) = AMAX                                                HYFIX
      HAREA(2,1,L+1) = APT(1,2,2)                                          HYFIX
      HAREA(1,2,L+1) = AMAX                                                HYFIX
      HAREA(2,2,L+1) = APT(2,2,2)                                          HYFIX
      IF (APT(2,2,2).GE.APT(1,2,2)) L = L + 1                              HYFIX
      HAREA(1,1,L+1) = APT(2,2,1)                                          HYFIX
      HAREA(2,1,L+1) = BMAX                                                HYFIX
      HAREA(1,2,L+1) = APT(1,2,1)                                          HYFIX
      HAREA(2,2,L+1) = BMAX                                                HYFIX
      IF (APT(2,2,1).GE.APT(1,2,1)) L = L + 1                              HYFIX
      IF (L.LE.1) GO TO 140                                                HYFIX
      HAREA(1,1,L+1) = HAREA(1,1,1)                                        HYFIX
      HAREA(2,1,L+1) = HAREA(2,1,1)                                        HYFIX
      IF (BD(1).GE.-2) CALL PLREA(L,HAREA,AREA,AB,BB,E,DELT,AMR)           HYFIX
      IF (BD(1).LT.-2) CALL HYREA(L,HAREA,AREA,AB,BB)                      HYFIX
      AREAL = AREA.GT.0.0                                                  HYFIX
      IF (.NOT.AREAL) GO TO 140                                            HYPER
C                                                                          HYPER
      DO 120 I = 1,3                                                       HYPER
      RM(I) = AB*U(I) + BB*V(I) + T4(I)                                    HYPER
  120 RMD(I) = RM(I)                                                       HYPER
COMPUTE POINT ON ELLIPSOID BELOW CENTROID       (CONTACT POINT?)           PLEDG
CONVERT PLANE VECTOR, ET = E*TM                                            PLEDG
C TRY TO USE OTHER LOGIC                                                   HYFIX
      IF(BD(1).LT.0.0)GO TO 130                                            HYPER
      CALL MAT31(BD(7),TM,ET)                                              PLEDG
      A2 = TM(1)*ET(1) + TM(2)*ET(2) + TM(3)*ET(3)                         PLEDG
      A1 = AB*(TM(1)*EU(1)+TM(2)*EU(2)+TM(3)*EU(3))                        HYFIX
     1+FM+ BB*(TM(1)*EV(1)+TM(2)*EV(2)+TM(3)*EV(3))                        HYFIX
      A1 = A1/A2                                                           HYFIX
      A0 = (AB**2*E(1,1) + 2.*AB*BB*E(1,2) + BB**2*E(2,2) - AMR)/A2        HYFIX
      DISC = A1**2 - A0                                                    PLEDG
      IF(DISC.LT.0.0)DISC = 0.0                                            PLEDG
      P = A1 + DSQRT(DISC)                                                 PLEDG
      GO TO 140                                                            HYPER
COMPUTE FOR HYPER                                                          HYPER
  130 CALL HYVAL(CA,TH,RM,BD,1)                                            HYFIX
      P = -CA                                                              HYFIX
      CALL DOT31(BD(8),RMD,RM)                                             HYPER
  140 RETURN                                                               HYPER
      END                                                                  PLEDG
```

```
      SUBROUTINE PLELP(M,MM,N,NN,NT)                                    PLELP
C                                                       REV IV    02/07/87HYPER
      IMPLICIT REAL*8(A-H,O-Z)                                          PLELP
      LOGICAL AREAL                                                     EDGE
      COMMON/TABLES/MXNTI,MXNTB,MXTB1,MXTB2,NTI(50),NTAB(1250),TAB(4500)PLELP
      COMMON/SGMNTS/ D(3,3,30),WMEG(3,30),WMEGD(3,30),U1(3,30),U2(3,30),PLELP
     *               SEGLP(3,30),SEGLV(3,30),SEGLA(3,30),NSYM(30)       PLELP
      COMMON/FORCES/PSF(7,70),BSF(4,20),SSF(10,40),BAGSF(3,20),         NCFORC
     *               PRJNT(7,30),NPANEL(5),NPSF,NBSF,NSSF,NBGSF         PLELP
      COMMON/CNTSRF/ PL(24,30),BELT(20,8),TPTS(6,8),BD(24,40)           EDGE
      COMMON/CSTRNT/ A13(3,3,24),A23(3,3,24),B31(3,3,24),B32(3,3,24),   PLELP
     *               HHT(3,3,12),RK1(3,12),RK2(3,12),QQ(3,12),TQQ(3,12),PLELP
     *               RQQ(3,12),HQQ(3,12),SQQ(12),CFQQ(12),              PLELP
     *               KQ1(12),KQ2(12),KQTYPE(12)                         PLELP
      COMMON/RSAVE/  XSG(3,20,3),DPMI(3,3,30),LPMI(30),                 TGMOD7
     *               NSG(9),MSG(20,9),MCG,MCGIN(24,5),KREF(20,9)        TGMOD7
      COMMON/TEMPVS/DMNT(3,3),TEMP(3,3),B(3,3),XMN(3),RLN(3),XMM(3),    PLELP
     *            TM(3),R(3),RM(3),DMNWN(3),RLM(3),RN(3),VMN(3),VR(3),  PLELP
     *            WNM(3),WCM(3),WCN(3),VREL(3),FFM(3),FR(3),TQM(3),     PLELP
     *            TQN(3),TQNT(3),T(3),H(3),TH(3),XH(3),RMD(3),RND(3),   EDGE
     *            TD(3),TT4(3,4),TT5(3,4),XNC(3),UH(3),P,AMR,FM,CF,     EDGE
     *            VRM,VRT,VRTS,VRTEST,TF,ELOSS,MCF,NCF                  TGMOD7
      CALL ELTIME(1,21)                                                 PLELP
      CALL DOTT33(D(1,1,M),D(1,1,N),DMNT)                               PLELP
      DO 10 I = 1,3                                                     PLELP
   10 XMN(I) = SEGLP(I,M) - SEGLP(I,N)                                  PLELP
      CALL MAT31(D(1,1,M),XMN,XMM)                                      PLELP
      CALL MAT31(DMNT,PL(1,NN),TM)                                      PLELP
      CALL MAT31(DMNT,PL(5,NN),TD)                                      EDGE
      BET = 0.0                                                         EDGE
      J = 3                                                             HYPER
      IF(BD(1,MM).LT.0.0) J = 4                                         HYPER
      DO 15 I=1,3                                                       EDGE
      J = J + 1                                                         HYPER
      XNC(I) = XMM(I) + BD(J,MM) - TD(I)                                HYPER
   15 BET = BET - TM(I)*XNC(I)                                          EDGE
C                                                                       EDGE
C     BET IS FROM CENTER OF FIGURE TO PLANE                             EDGE
      IF(BD(1,MM).GT.0.0)GO TO 30                                       HYPER
C PUT PLANE VECTOR INTO HYPER                                           HYPER
      CALL MAT31(BD(8,MM),TM,TH)                                        HYPER
      CALL MAT31(BD(8,MM),XNC,UH)                                       HYPER
      DO 20 I = 1,3                                                     HYPER
      XNC(I) = UH(I)                                                    HYPER
      UH(I) = DABS(TH(I))*BD(I+1,MM)/BD(I+19,MM)                        HYPER
      R(I)  = BD(I+19,MM)/(BD(I+19,MM) - 1.0)                           HYPER
   20 RND(I)= UH(I)**R(I)                                               HYPER
      ALP = HYPEN(BD(1,MM),R,RND)                                       HYPER
      DO 25 I = 1,3                                                     HYPER
      POW = 1.0/(BD(I+19,MM) - 1.0)                                     HYPER
```

```
      XH(I) = -DSIGN(BD(I+1,MM)*(UH(I)*ALP)**POW,TH(I))                 HYPER
   25 RND(I) = XH(I)                                                     HYPER
      BTE = TH(1)*XH(1) + TH(2)*XH(2) + TH(3)*XH(3)                      HYPER
      FM = BET/BTE                                                       HYPER
      AMR = 1.0 - DABS(FM)**(-BD(1,MM))                                  HYPER
      GO TO 35                                                           HYPER
C     CODE FOR ELLIPSE         XH = E'T                                  EDGE
C                                                                        EDGE
   30 CALL MAT31(BD(16,MM),TM,XH)                                        HYPER
      BTS = TM(1)*XH(1) + TM(2)*XH(2) + TM(3)*XH(3)                      EDGE
      BTE = - DSQRT(BTS)                                                 PLELP
      FM = BET/BTS                                                       EDGE
      AMR = 1.0 -BET*FM                                                  EDGE
C                                                                        EDGE
   35 P = BET - BTE                                                      HYPER
      PSF(1,NPSF) = P                                                    PLELP
      MCF = NTAB(NT+1)                                                   PLELP
      NCF = -MCF                                                         PLELP
      IF(NCF.GT.0)CFQQ(NCF) = -999.                                      PLELP
      IF(P.LE.0.0) GO TO 85                                              HYPER
C                                                                        EDGE
C     CALL EDGE ROUTINE TO FIND IF ELLIPSOID INTERSECTS FINITE PLANE     EDGE
C     IF IT DOES; AREAL WILL BE TRUE, P WILL BE PENETRATION AT CENTROID EDGE
C     AND RM WILL BE LOCATION OF CENTROID                                EDGE
C     RM IS REFERENCED TO CENTER OF ELLIPSOID                            EDGE
C     USE OLD FORMULA FOR ROLL-SLIDE?, I.E. ROLL-SLIDE SHOULDN'T         EDGE
C     CALL PLEDG                                                         EDGE
C                                                                        EDGE
      LT = NTAB(NT)                                                      EDGE
      IF(TAB(LT+22).LE.0.0)GO TO 40                                      HYPER
C                                                                        EDGE
      IF (AMR.LE.0.0) GO TO 85                                           HYPER
      IF (BD(1,MM).LT.0.0.AND.BD(23,MM).NE.0.0) STOP 22                  HYPER
      CALL PLEDG(AREAL,BD(1,MM),PL(1,NN))                                EDGE
      IF(.NOT.AREAL)GO TO 85                                             HYPER
      PSF(1,NPSF) = P                                                    EDGE
C                                                                        EDGE
   40 IF (TAB(LT+22).GT.-2.0.AND.AMR.LE.0.0) GO TO 85                    HYPER
      RHO = 0.0                                                          HYPER
      IF(MCF.GT.0)RHO = TAB(MCF+4)                                       PLELP
      BETE =  1.0 + RHO*P/BTE                                            HYPER
      IF(BD(1,MM).GT.0.0)BETE = BETE/BTE                                 HYPER
      IF(BD(1,MM).LT.0.0)CALL DOT31(BD(8,MM),RND,XH)                     HYPER
      TRT = P*(1.0 - RHO)                                                EDGE
      J = 3                                                              HYPER
      IF(BD(1,MM).LT.0.0)J = 4                                           HYPER
      DO 45 I = 1,3                                                      HYPER
      J = J + 1                                                          HYPER
      IF(TAB(LT+22).LE.0.0)RM(I) = BETE*XH(I)                            EDGE
      IF(TAB(LT+22).GT.0.0)RM(I) = RM(I) - TRT*TM(I)                     EDGE
```

```
      RLM(I) = RM(I) + BD(J,MM)                                       HYPER
   45 RN(I) = RLM(I) + XMM(I)                                         HYPER
      CALL DOT31(DMNT,RN,RLN)                                         PLELP
      IF (TAB(LT+22).GT.0.0) GO TO 55                                 HYPER
      IF (TAB(LT+22).GT.-3.0.AND.TAB(LT+22).LT.0.0) GO TO 55          HYPER
C                                                                     EDGE
C     CHECK BOUNDARY USING OLD METHOD                                 EDGE
      DO 50 I = 8,13,5                                                HYPER
      IF(PL(I+4,NN).LE.0.0)GO TO 50                                   HYPER
      DIST = RLN(1)*PL(I   ,NN)                                       PLELP
     *       + RLN(2)*PL(I+1,NN)                                      PLELP
     *       + RLN(3)*PL(I+2,NN) - PL(I+3,NN)                         PLELP
      IF((DIST.LE.0.0).OR.(DIST.GT.PL(I+4,NN))) GO TO 85              HYPER
   50 CONTINUE                                                        HYPER
C                                                                     EDGE
   55 CALL PLSEGF(M,N,NT)                                             HYPER
C     DMNWN,VMN,VR,WNM,WCM,WCN,VREL,FFM,FR,TQM,TQN,TQNT,T             EDGE
C     FM,CF,VRM,VRT,VRTS,VRTEST,TF,ELOSS                              EDGE
C                                                                     EDGE
C     STORE RESULTS                                                   EDGE
      DO 60 I = 1,3                                                   HYPER
   60 PSF(I+4,NPSF) = RLN(I)                                          HYPER
      IF(LPMI(N).NE.0) CALL DOT31(DPMI(1,1,N),RLN,PSF(5,NPSF))        EDGE
      IF(MCF.LT.0)GO TO 65                                            HYPER
      PSF(2,NPSF) = FM                                                PLELP
      PSF(3,NPSF) = 0.0                                               PLELP
      TRT = TF**2 - FM**2                                             PLELP
      IF(TRT.GT.0.0) PSF(3,NPSF) = DSQRT(TRT)                         PLELP
      PSF(4,NPSF) = TF                                                PLELP
      GO TO 85                                                        HYPER
C                                                                     PLELP
C      ROLL-SLIDE             REVISED          8/18/85                PLELP
   65 DO 70 I = 1,3                                                   HYPER
   70 PSF(I+1,NPSF) = T(I)                                            HYPER
      IF(BD(1,MM).LT.0.0) STOP 28                                     HYPER
      CALL CROSS(TM,WNM,TH)                                           EDGE
      CALL MAT31(BD(16,MM),TH,UH)                                     EDGE
      TRT = (TM(1)*UH(1) + TM(2)*UH(2) + TM(3)*UH(3))/BTS             EDGE
      DO 75 I = 1,3                                                   HYPER
   75 RMD(I) = DABS(BETE)*(UH(I) - TRT*XH(I))                         HYPER
      CALL CROSS(DMNWN,TM,TH)                                         EDGE
      CALL CROSS(WNM,RMD,XNC)                                         EDGE
      SQQ(NCF) = 0.0                                                  PLELP
      DO 80 I = 1,3                                                   HYPER
   80 SQQ(NCF) = SQQ(NCF) + TM(I)*XNC(I) - 2.0*TH(I)*VR(I)            HYPER
      CALL DOT31(D(1,1,M),XNC,RQQ(1,NCF))                             EDGE
   85 CALL ELTIME(2,21)                                               HYPER
      RETURN                                                          PLELP
      END                                                             PLELP
```

```
      SUBROUTINE PLREA(L,H,AREA,AB,BB,E,D,R)                              HYFIX
C                                                             REV IV 12/11/87HYFIX
      IMPLICIT REAL*8(A-H,O-Z)                                            PLREA
COMPUTES AREA AND CENTROID (TRUE AREA = AREA*|UxV|/6)                     HYFIX
C |UxV| IS NEVER COMPUTED    |UxV| = UxV.T = AREA OF PARALLELOGRAM        PLREA
C THIS ROUTINE WILL ONLY BE CALLED IF THERE IS AN INTERSECTION            PLREA
      DIMENSION H(2,2,5),E(2,2)                                           HYFIX
      AREA = 0.0                                                          PLREA
      AB = 0.0                                                            PLREA
      BB = 0.0                                                            PLREA
      IF (L.LE.1) GO TO 15                                                HYFIX
      C = R/DSQRT(D)                                                      HYFIX
      C12 = 2.0*R/D                                                       HYFIX
      C11 = C12*E(1,1)                                                    HYFIX
      C22 = C12*E(2,2)                                                    HYFIX
      C12 = C12*E(1,2)                                                    HYFIX
      DO 10 I = 1,L                                                       HYFIX
COMPUTE FOR STRAIGHT LINE SEGMENTS                                        HYFIX
      AR = H(1,1,I)*H(2,2,I) - H(1,2,I)*H(2,1,I)                          HYFIX
      IF (AR.EQ.0.0) GO TO 5                                              HYFIX
      AB = AB + AR*(H(1,1,I) + H(1,2,I))                                  HYFIX
      BB = BB + AR*(H(2,1,I) + H(2,2,I))                                  HYFIX
      AREA = AREA + AR                                                    HYFIX
COMPUTE FOR ELLIPSE                                                       HYFIX
    5 AR = H(1,2,I)*H(2,1,I+1) - H(1,1,I+1)*H(2,2,I)                      HYFIX
      IF (AR.EQ.0.0) GO TO 10                                             HYFIX
      ARC = AR/C                                                          HYFIX
      IF (DABS(ARC).GT.1.0) ARC = DSIGN(1.0D0,ARC)                        HYFIX
      AR = C*DASIN(ARC)                                                   HYFIX
      X21 = H(1,1,I+1) - H(1,2,I)                                         HYFIX
      Y21 = H(2,1,I+1) - H(2,2,I)                                         HYFIX
      AB = AB + C12*X21 + C22*Y21                                         HYFIX
      BB = BB - C11*X21 - C12*Y21                                         HYFIX
      AREA = AREA + AR                                                    HYFIX
   10 CONTINUE                                                            HYFIX
      IF (AREA.LE.0.0) GO TO 15                                           HYFIX
      AREA = 3.0*AREA                                                     HYFIX
      AB = AB/AREA                                                        PLREA
      BB = BB/AREA                                                        PLREA
C     AREA = AREA/6.0                                                     HYFIX
   15 RETURN                                                              PLREA
      END                                                                 PLREA
```

```
      SUBROUTINE PLSEGF(M,N,NT)                                         PLSEGF
C                                                 REV III.5 09/03/85TGMOD7
      IMPLICIT REAL*8 (A-H,O-Z)                                         PLSEGF
      COMMON/SGMNTS/ D(3,3,30),WMEG(3,30),WMEGD(3,30),U1(3,30),U2(3,30),PLSEGF
     *               SEGLP(3,30),SEGLV(3,30),SEGLA(3,30),NSYM(30)       PLSEGF
      COMMON/CSTRNT/ A13(3,3,24),A23(3,3,24),B31(3,3,24),B32(3,3,24),   PLSEGF
     *               HHT(3,3,12),RK1(3,12),RK2(3,12),QQ(3,12),TQQ(3,12),PLSEGF
     *               RQQ(3,12),HQQ(3,12),SQQ(12),CFQQ(12),              PLSEGF
     *               KQ1(12),KQ2(12),KQTYPE(12)                         PLSEGF
      COMMON/TEMPVI/ CREST,TTI(3),R1I(3),R2I(3),JSTOP(4,2,30)           PLSEGF
      COMMON/TABLES/MXNTI,MXNTB,MXTB1,MXTB2,NTI(50),NTAB(1250),TAB(4500)DIMENB
C     THIS COMMON/TEMPVS/ IS SHARED BY PLELP, PLSEGF AND SEGSEG.         PLSEGF
      COMMON/TEMPVS/DMNT(3,3),TEMP(3,3),B(3,3),XMN(3),RLN(3),XMM(3),    PLSEGF
     *              TM(3),R(3),RM(3),DMNWN(3),RLM(3),RN(3),VMN(3),VR(3),PLSEGF
     *              WMN(3),WCM(3),WCN(3),VREL(3),FFM(3),FR(3),TQM(3),   PLSEGF
     *              TQN(3),TQNT(3),T(3),H(3),T1(3),T2(3),RMD(3),RND(3), PLSEGF
     *              TD(3),TT4(3,4),TT5(3,4),T3(3),T4(3),P,AMR,FM,CF,    PLSEGF
     *              VRM,VRT,VRTS,VRTEST,TF,ELOSS,MCF,NCF,T5(3),T6(3)    TGMOD7
      VRTEST = 2.0                                                      PLSEGF
      CALL MAT31(DMNT,WMEG(1,N),DMNWN)                                  PLSEGF
      DO 15 I=1,3                                                       PLSEGF
      VMN(I) = SEGLV(I,M) - SEGLV(I,N)                                  PLSEGF
   15 WMN(I) = DMNWN(I) - WMEG(I,M)                                     PLSEGF
      CALL DOT31(D(1,1,M),TM,T)                                         PLSEGF
      CALL MAT31(D(1,1,M),VMN,VR)                                       PLSEGF
      CALL CROSS(WMEG(1,M),RLM,WCM)                                     PLSEGF
      CALL CROSS(DMNWN,RN,WCN)                                          PLSEGF
      VRM = 0.0                                                         PLSEGF
      DO 16 I=1,3                                                       PLSEGF
      VR(I) = VR(I) + WCM(I) - WCN(I)                                   PLSEGF
   16 VRM = VRM + VR(I)*TM(I)                                           PLSEGF
      VRT = 0.0                                                         PLSEGF
      DO 17 I=1,3                                                       PLSEGF
      VREL(I) = VR(I) - VRM*TM(I)                                       PLSEGF
   17 VRT = VRT + VREL(I)**2                                            PLSEGF
      VRT = DSQRT(VRT)                                                  PLSEGF
      CF = EVALFD (P,NTAB(NT+5),1)                                      PLSEGF
      LT = NTAB(NT)                                                     PLSEGF
      TAB(LT) = P                                                       PLSEGF
      FM = 1.0                                                          PLSEGF
      PDOT = -VRM                                                       PLSEGF
      ELOSS = 0.0                                                       PLSEGF
      IF (MCF.GT.0) CALL FRCDFL(P,PDOT,NT,1,FM,ELOSS)                   PLSEGF
      VRTS = VRT                                                        PLSEGF
      IF (VRT.LT.VRTEST) VRT = VRTEST/(2.0-VRT/VRTEST)                  PLSEGF
      FF = -DABS(FM)*CF/VRT                                             PLSEGF
      IF (NCF.GT.0.AND.KQTYPE(NCF).EQ.6)  FF=0.0                        PLSEGF
      FS = (VRTS-VRT)/VRT                                               PLSEGF
      IF (NCF.GT.0.AND.KQTYPE(NCF).EQ.6)  FS=0.0                        PLSEGF
      TF = 0.0                                                          PLSEGF
```

```
      L = LT+18                                                  PLSEGF
      DO 18 I=1,3                                                PLSEGF
      L = L+1                                                    PLSEGF
      FFM(I) = FM*TM(I) + FF*VREL(I) + FS*TAB(L)                 PLSEGF
      TF = TF + FFM(I)**2                                        PLSEGF
      TTI(I) = T(I)                                              PLSEGF
      R1I(I) = RLM(I)                                            PLSEGF
   18 R2I(I) = RLN(I)                                            PLSEGF
      TF = DSQRT(TF)                                             PLSEGF
      MT = NTAB(NT+5)                                            PLSEGF
      CREST = TAB(MT+3)                                          PLSEGF
      CALL DOT31 (D(1,1,M),FFM,FR)                               PLSEGF
      IF (MCF.LE.0) GO TO 21                                     PLSEGF
      CALL CROSS (RLM,FFM,TQM)                                   PLSEGF
      CALL CROSS (RN,FFM,TQNT)                                   PLSEGF
      CALL DOT31 (DMNT,TQNT,TQN)                                 PLSEGF
      DO 19 I=1,3                                                PLSEGF
      U1(I,M) = U1(I,M) + FR(I)                                  PLSEGF
      U1(I,N) = U1(I,N) - FR(I)                                  PLSEGF
      U2(I,M) = U2(I,M) + TQM(I)                                 PLSEGF
   19 U2(I,N) = U2(I,N) - TQN(I)                                 PLSEGF
      IF (NCF.LE.0) GO TO 23                                     PLSEGF
   21 DO 22 I=1,3                                                PLSEGF
      HQQ(I,NCF) = FR(I)/TF                                      PLSEGF
      TQQ(I,NCF) = T(I)                                          PLSEGF
      RK1(I,NCF) = RLM(I)                                        PLSEGF
   22 RK2(I,NCF) = RLN(I)                                        PLSEGF
      CFQQ(NCF)  = CF                                            PLSEGF
      MT = NTAB(NT+5)                                            PLSEGF
      IF (KQTYPE(NCF).EQ.3) CFQQ(NCF) = TAB(MT+4)                PLSEGF
   23 RETURN                                                     PLSEGF
      END                                                        PLSEGF
```

```
      SUBROUTINE PLTXYZ(P,C)                                            PLTXYZ
C                                                  REV III.5 05/30/85VEHICL
C     STORES PLOT CHARACTER (C) INTO PLOTYZ, PLOTXZ AND PLOTXY ARRAYS   PLTXYZ
C     IN VEHICLE REFERENCE FOR POINT (P) GIVEN IN INERTIAL REFERENCE.   PLTXYZ
C                                                                       PLTXYZ
      IMPLICIT REAL*8 (A-H,O-Z)                                         PLTXYZ
      COMMON/CONTRL/ TIME,NSEG,NJNT,NPL,NBLT,NBAG,NVEH,NGRND,           PLTXYZ
     *               NS,NQ,NSD,NFLX,NHRNSS,NWINDF,NJNTF,NPRT(36),NPG    PAGE
      COMMON/SGMNTS/ D(3,3,30),WMEG(3,30),WMEGD(3,30),U1(3,30),U2(3,30),PLTXYZ
     *               SEGLP(3,30),SEGLV(3,30),SEGLA(3,30),NSYM(30)       PLTXYZ
      COMMON/VPOSTN/ ZPLT(3),SPLT(3),AXV(3,6),VATAB(6,501,6),           VEHICL
     *               VTO(6),VDT(6),TIMEV(6),OMEGV(6),NVTAB(6),INDXV(6)  PLTXYZ
      COMMON/TEMPVS/ DUM(101),PLOTYZ(96,55),PLOTXZ(96,55),PLOTXY(96,55) PLTXYZ
      LOGICAL*1   C,PLOTYZ,PLOTXZ,PLOTXY                                PLTXYZ
      DIMENSION P(3),TMP(3),XYZ(3)                                      PLTXYZ
      DATA NPLTZ/96/ , NPLTX/55/                                        PLTXYZ
C                                                                       PLTXYZ
C     CONVERT P FROM INERTIAL TO VEHICLE REFERENCE BY                   PLTXYZ
C         XYZ = DVEH(P-XCOMP)                                           PLTXYZ
C                                                                       PLTXYZ
      DO 10 I=1,3                                                       PLTXYZ
   10 TMP(I) = P(I) - SEGLP(I,NVEH)                                     PLTXYZ
      CALL MAT31(D(1,1,NVEH),TMP,XYZ)                                   PLTXYZ
C                                                                       PLTXYZ
C     CONVERT XYZ INTO PLOT CORDINATES IX,IY,IZ AND                     PLTXYZ
C     IF WITHIN PLOT LIMITS, STORE C IN PLOTYZ, PLOTXZ AND PLOTXY.      PLTXYZ
C                                                                       PLTXYZ
      IX = SPLT(1)*XYZ(1) + ZPLT(1) + 0.5                               PLTXYZ
      IZ = SPLT(3)*XYZ(3) + ZPLT(3) + 0.5                               PLTXYZ
      IF (IZ.LT.1 .OR. IZ.GT.NPLTZ) GO TO 11                            PLTXYZ
      IY = SPLT(2)*XYZ(2) + ZPLT(2) + 0.5                               PLTXYZ
      IF (IY.GE.1 .AND. IY.LE.NPLTX) PLOTYZ(IZ,IY) = C                  PLTXYZ
      IF (IX.GE.1 .AND. IX.LE.NPLTX) PLOTXZ(IZ,IX) = C                  PLTXYZ
   11 IY = -SPLT(3)*XYZ(2) + ZPLT(2) + 0.5                              PLTXYZ
      IF (IY.LT.1 .OR.  IY.GT.NPLTZ) GO TO 99                           PLTXYZ
      IF (IX.GE.1 .AND. IX.LE.NPLTX) PLOTXY(IY,IX) = C                  PLTXYZ
   99 RETURN                                                            PLTXYZ
      END                                                               PLTXYZ
```

```
      SUBROUTINE POSTPR(PRDT)                                           POSTPR
C                                                    REV IV    02/01/88MISDOT
C     CONTROLS GENERATION OF PRINTED TABULAR TIME HISTORIES             POSTPR
C     AND PLOTS BY THE VALUE OF NPRT(4) AS FOLLOWS:                     POSTPR
C                                                                       POSTPR
C      VALUE OF      TIME                                               POSTPR
C      NPRT(4)    HISTORIES   PLOTS                                     POSTPR
C                                                                       POSTPR
C        +4          **        NO                                       POSTPR
C        +3         YES       YES                                       POSTPR
C        +2         YES        NO                                       POSTPR
C        +1          **       YES                                       POSTPR
C         0          **        NO                                       POSTPR
C        -1          NO       YES                                       POSTPR
C        -2         YES        NO                                       POSTPR
C        -3         YES       YES                                       POSTPR
C                                                                       POSTPR
C      **  TIME HISTORIES WERE PRINTED BY SUBROUTINE OUTPUT.            POSTPR
C                                                                       POSTPR
      COMMON/CDINT/ JDTPTS(18),ZZ(1000,3)                               PLTINC
C     NOTE: THIS OVERWRITES COMMON /CDINT/.                             POSTPR
      COMMON/CONTRL/ TIME,NSEG,NJNT,NPL,NBLT,NBAG,NVEH,NGRND,           POSTPR
     *               NS,NQ,NSD,NFLX,NHRNSS,NWINDF,NJNTF,NPRT(36),NPG    PAGE
      REAL*8 TIME                                                       POSTPR
      COMMON/FORCES/PSF(7,70),BSF(4,20),SSF(10,40),BAGSF(3,20),         NCFORC
     *               PRJNT(7,30),NPANEL(5),NPSF,NBSF,NSSF,NBGSF         POSTPR
      REAL*8 PSF,BSF,SSF,BAGSF,PRJNT                                    POSTPR
      COMMON/TITLES/ DATE(3),COMENT(40),VPSTTL(20),BDYTTL(5),           POSTPR
     *               BLTTTL(5,8),PLTTL(5,30),BAGTTL(5,6),SEG(30),       POSTPR
     *               JOINT(30),CGS(30),JS(30)                           POSTPR
      REAL DATE,COMENT,VPSTTL,BDYTTL,BLTTTL,PLTTL,BAGTTL,SEG,JOINT      POSTPR
      LOGICAL*1 CGS,JS                                                  POSTPR
      COMMON/CNSNTS/ PI,RADIAN,G,THIRD,EPS(24),                         POSTPR
     *               UNITL,UNITM,UNITT,GRAVTY(3),TWOPI                  TWOPI
      REAL*8 PI,RADIAN,G,THIRD,EPS,UNITL,UNITM,UNITT,GRAVTY             POSTPR
      COMMON/JBARTZ/ MNPL(   30),MNBLT(   8),MNSEG(   30),MNBAG(    6), POSTPR
     *               MPL(3,5,30),MBLT(3,5,8),MSEG(3,5,30),MBAG(3,10,6), POSTPR
     *               NTPL( 5,30),NTBLT( 5,8),NTSEG( 5,30)               POSTPR
      COMMON/DAMPER/ APSDM(3,20),APSDN(3,20),ASD(5,20),MSDM(20),MSDN(20)POSTPR
      REAL*8 APSDM,APSDN,ASD                                            POSTPR
      COMMON/HRNESS/ BAR(15,100),BB(100),BBDOT(100),PLOSS(2,100),       POSTPR
     *               XLONG(20),HTIME(2),IBAR(5,100),NL(2,100),          POSTPR
     *               NPTSPB(20),NPTPLY(20),NTHRNS(20),NBLTPH(5)         POSTPR
      REAL*8 BAR,BB,BBDOT,PLOSS,XLONG,HTIME                             POSTPR
      COMMON/RSAVE/  XSG(3,20,3),DPMI(3,3,30),LPMI(30),                 ATBIII
     *               NSG(9),MSG(20,9),MCG,MCGIN(24,5),KREF(20,9)        TTHKREF
      REAL*8 XSG,DPMI,TDATA,UMSEC,PRDT,TEST1,TEST2,VDT1                 TGMOD1
      REAL*8 VDT2,R30,R26                                               TGMOD1
C     NOTE: SUBROUTINES POSTPR & HEDING SHARE THIS COMMON/TEMPVS/.      POSTPR
C     THE FIRST DIMENSION OF XLAB,YLAB,PLB1 AND PLB2 SHOULD BE THE SAME POSTPR
```

```
C     AS THE VALUE ASSIGNED TO NW60 WHICH IS THE NUMBER OF WORDS THAT   POSTPR
C     IS NECESSARY TO CONTAIN 60 CONSECUTIVE CHARACTERS DEPENDING ON THEPOSTPR
C     COMPUTER SYSTEM THIS PROGRAM IS OPERATING ON. THE VALUE OF NW60   POSTPR
C     SHOULD BE 15 ON IBM 360 AND 370, 10 ON UNIVAC 1108, 6 ON CDC 6600.POSTPR
C     THE LAST TERM IN FORMAT 13 BELOW SHOULD BE 15A4(IBM), 10A6(UNIVAC)POSTPR
C     OR 6A10(CDC). ALSO, THE FIRST DIMENSION OF PLDATA IN SUBROUTINE   POSTPR
C     HEDING SHOULD BE 97(IBM), 77(UNIVAC) OR 61(CDC).                  REDIM2
C                                                                       POSTPR
      COMMON/TEMPVS/ TDATA(14,65),HEDATA(470),                          POSTPR
     *              X0(20),XN(20),XL(20),XS(20),XLAB(15,20),PLB1(15,20),POSTPR
     *              Y0(20),YN(20),YL(20),YS(20),YLAB(15,20),PLB2(15,20),POSTPR
     *              NYP(20),MX(2,20),MY(2,10,20),NX(20),NY(20),         POSTPR
     *              NXLAB(20),NYLAB(20),NPLB1(20),NPLB2(20),            POSTPR
     *              USEC(45),Z(1000,25),ZTTH(14,45,65)                  MISDOT
      LOGICAL LTABH,LPLOT                                               POSTPR
      DATA  LPP/45/ , NZD1/1000/, NZD2/25/                              PLTINC
      DATA NW60/15/                                                     POSTPR
      LTABH = .FALSE.                                                   POSTPR
      LPLOT = .FALSE.                                                   POSTPR
      NPRT4 = IABS(NPRT(4))                                             POSTPR
      LPLOT = NPRT4.EQ.1 .OR. NPRT4.EQ.3                                POSTPR
      LTABH = NPRT4.EQ.2 .OR. NPRT4.EQ.3                                POSTPR
      IF(NPRT(26).EQ.4) LTABH = .FALSE.                                 TGMOD1
      IF(NPRT(26).GE.5) GO TO 99                                        TGMOD1
C                                                                       POSTPR
C     READ INPUT CARD H.11 TO CONTROL COMPUTATION OF HIC, HSI & CSI.    WINDOP
C                                                                       POSTPR
      READ  (5,11)  JDTPTS                                              POSTPR
      WRITE(6,700) NPG                                                  PAGE
      NPG=NPG+1                                                         PAGE
  700 FORMAT(1H1,122X,'PAGE',I5/,2X,                                    PAGE
     *       'POSTPROCESSOR CONTROL PARAMETERS',/)                      PAGE
      WRITE(6,701)                                                      CHGIII
  701 FORMAT(13X,'HIC & HSI POINT',7X,'CSI POINT')                      CHGIII
      WRITE(6,702) JDTPTS(1),JDTPTS(2)                                  CHGIII
  702 FORMAT(5X,'H.11',10X,I2,17X,I2,//)                                WINDOP
      NDPT = 0                                                          POSTPR
      IHIC = 0                                                          TGMOD1
      I26 = 0                                                           TGMOD1
      ITST1 = 0                                                         TGMOD1
      ITST2 = 0                                                         TGMOD1
      IF(NPRT(26).LT.0) I26 = IABS(NPRT(26))                            TGMOD1
      IF(JDTPTS(1).GT.0.OR.JDTPTS(2).GT.0) IHIC = 1                     TGMOD1
      IF(NPRT(30).EQ.0.AND.NPRT(26).EQ.3) ITST1 = 1                     TGMOD1
      IF(NPRT(30).LT.I26) ITST2 = 1                                     TGMOD1
      IF(IHIC.EQ.1.AND.ITST1.EQ.1) WRITE(6,751)                         TGMOD1
      IF(IHIC.EQ.1.AND.ITST2.EQ.1) WRITE(6,752) NPRT(30),I26            TGMOD1
  751 FORMAT(3X,'WARNING!  LOGIC OF INPUT INDICATES USER ANTICIPATES HICTGMOD1
     *, HSI AND CSI TO BE COMPUTED BASED ON DATA FOR EVERY SUCCESSFUL', TGMOD1
     */,10X,'INTEGRATION STEP, YET DATA WAS STORED (WRITTEN TO TAPE8) EVTGMOD1
```

```
     *ERY DT.')                                                          TGMOD1
  752 FORMAT(3X,'WARNING!  LOGIC OF INPUT INDICATES USER ANTICIPATES HICTGMOD1
     *, HSI AND CSI TO BE COMPUTED BASED ON DATA FOR EVERY ',I2,/,10X,' TGMOD1
     *INTEGER MULTIPLE OF DT, YET DATA WAS STORED (WRITTEN TO TAPE8) EVETGMOD1
     *RY ',I2,' INTEGER MULTIPLE OF DT.')                                TGMOD1
      IF(JDTPTS(1).GT.0.AND.NPRT(26).EQ.2.AND.NPRT(30).LT.1) STOP 91     TGMOD1
      IF(JDTPTS(2).GT.0.AND.NPRT(26).EQ.2.AND.NPRT(30).LT.1) STOP 92     TGMOD1
      IF  (JDTPTS(1).NE.0)  NDPT = NDPT + 1                              POSTPR
      IF  (JDTPTS(2).NE.0)  NDPT = NDPT + 1                              POSTPR
      IF  (.NOT.LPLOT .AND. .NOT.LTABH .AND. NDPT.EQ.0) GO TO 99         POSTPR
      CALL ELTIME (1,36)                                                 POSTPR
      IF  (.NOT.LPLOT)  GO TO 20                                         POSTPR
C                                                                        POSTPR
C     READ INDICES OF VARIABLES TO BE PLOTTED AND                        POSTPR
C     ARGUMENTS TO SUBROUTINE SLPLOT ON CARDS I.                         POSTPR
C                                                                        POSTPR
C         INPUT CARD I.1                                                 POSTPR
C                                                                        POSTPR
      READ  (5,11)  NPLT , (NYP(K),K=1,NPLT)                             POSTPR
   11 FORMAT (18I4)                                                      POSTPR
      IF(NPLT.GT.0.AND.ITST1.EQ.1) WRITE(6,753)                          TGMOD1
      IF(NPLT.GT.0.AND.ITST2.EQ.1) WRITE(6,754) NPRT(30),I26             TGMOD1
  753 FORMAT(3X,'WARNING!  LOGIC OF INPUT INDICATES USER ANTICIPATES PLOTGMOD1
     *TS TO BE COMPUTED BASED ON DATA FOR EVERY SUCCESSFUL INTEGRATION STGMOD1
     *TEP',/,10X,'YET DATA WAS STORED (WRITTEN TO TAPE8) EVERY DT.')     TGMOD1
  754 FORMAT(3X,'WARNING!  LOGIC OF INPUT INDICATES USER ANTICIPATES PLOTGMOD1
     *TS TO BE COMPUTED BASED ON DATA FOR EVERY ',I2,/,10X,'INTEGER MULTTGMOD1
     *PLE OF DT, YET DATA WAS STORED (WRITTEN TO TAPE8) EVERY ',I2,      TGMOD1
     *     ' INTEGER MULTIPLE OF DT.')                                   TGMOD1
      IF  (NPLT.LE.0)  LPLOT = .FALSE.                                   POSTPR
      IF  (.NOT.LPLOT)  GO TO 20                                         POSTPR
      DO  15  K=1,NPLT                                                   POSTPR
      NYPLT = NYP(K)                                                     POSTPR
C                                                                        POSTPR
C         INPUT CARD I.2.K                                               POSTPR
C                                                                        POSTPR
      READ  (5,11)  MX(1,K), MX(2,K), (MY(1,J,K), MY(2,J,K), J=1,NYPLT) POSTPR
C                                                                        POSTPR
C         INPUT CARD I.3.K                                               POSTPR
C                                                                        POSTPR
      READ  (5,12)  NX(K), X0(K), XN(K), XL(K), XS(K)                    POSTPR
   12 FORMAT (I4 , 4X , 4F8.0 )                                          POSTPR
C                                                                        POSTPR
C         INPUT CARD I.4.K                                               POSTPR
C                                                                        POSTPR
      READ  (5,12)  NY(K), Y0(K), YN(K), YL(K), YS(K)                    POSTPR
C                                                                        POSTPR
C         INPUT CARD I.5.K                                               POSTPR
C                                                                        POSTPR
      READ  (5,13)  NXLAB(K), (XLAB(I,K),I=1,NW60)                       POSTPR
```

```
  13  FORMAT (I4 , 4X , 15A4)                                            POSTPR
C                                                                        POSTPR
C     NOTE - ABOVE FORMAT ASSUMES 4 ALPHANUMERIC CHARACTERS FOR SINGLE   POSTPR
C     PRECISION WORDS ON IBM 360 AND 370 COMPUTERS. THE 15A4 TERM IN THEPOSTPR
C     FORMAT WILL HAVE TO BE CHANGED ON NON-IBM COMPUTERS TO PRODUCE A   POSTPR
C     CONTINUOUS STRING OF 60 CHARACTERS IN CORE MEMORY.                 POSTPR
C                                                                        POSTPR
C         INPUT CARD I.6.K                                               POSTPR
C                                                                        POSTPR
      READ  (5,13)  NYLAB(K), (YLAB(I,K),I=1,NW60)                       POSTPR
C                                                                        POSTPR
C         INPUT CARD I.7.K                                               POSTPR
C                                                                        POSTPR
      READ  (5,13)  NPLB1(K), (PLB1(I,K),I=1,NW60)                       POSTPR
C                                                                        POSTPR
C         INPUT CARD I.8.K                                               POSTPR
C                                                                        POSTPR
  15  READ  (5,13)  NPLB2(K), (PLB2(I,K),I=1,NW60)                       POSTPR
C                                                                        CHGIII
C     WRITE OUT PLOTTING CONTROL DATA                                    CHGIII
C                                                                        CHGIII
      WRITE(6,703)                                                       CHGIII
  703 FORMAT(4X,'PLOTTING CONTROLS',/)                                   CHGIII
      WRITE(6,704)                                                       CHGIII
  704 FORMAT(12X,'NO. PLOTS',11X,'NO. OF Y VARIABLES PER PLOT')          CHGIII
      WRITE(6,705) NPLT,(NYP(JK),JK=1,NPLT)                              CHGIII
  705 FORMAT(5X,'I.1',7X,I2,7X,20(I2,2X))                                CHGIII
      WRITE(6,706)                                                       CHGIII
  706 FORMAT(12X,'MX1 MX2 MY1A MY2A MY1B MY2B MY1C MY2C MY1D MY2D MY1E MCHGIII
     *Y2E MY1F MY2F MY1G MY2G MY1H MY2H MY1I MY2I MY1J MY2J')            CHGIII
      DO 730 IJ=1,NPLT                                                   CHGIII
      WRITE(6,707) IJ,MX(1,IJ),MX(2,IJ),                                 CHGIII
     * (MY(1,L,IJ),MY(2,L,IJ),L=1,NYP(IJ))                               CHGIII
  707 FORMAT(5X,'I.2.',I2,2X,I2,2X,I2,2X,20(I2,3X))                      CHGIII
  730 CONTINUE                                                           CHGIII
      WRITE(6,708)                                                       CHGIII
  708 FORMAT(14X,'NX',8X,'X0',9X,'XN',8X,'XL',9X,'XS')                   CHGIII
      DO 731 IJ=1,NPLT                                                   CHGIII
      WRITE(6,709) IJ,NX(IJ),X0(IJ),XN(IJ),XL(IJ),XS(IJ)                 CHGIII
  709 FORMAT(5X,'I.3.',I2,2X,I3,4X,4(F8.3,2X))                           CHGIII
  731 CONTINUE                                                           CHGIII
      WRITE(6,710)                                                       CHGIII
  710 FORMAT(14X,'NY',8X,'Y0',9X,'YN',8X,'YL',9X,'YS')                   CHGIII
      DO 732 IJ=1,NPLT                                                   CHGIII
      WRITE(6,711) IJ,NY(IJ),Y0(IJ),YN(IJ),YL(IJ),YS(IJ)                 CHGIII
  711 FORMAT(5X,'I.4.',I2,2X,I3,4X,4(F8.3,2X))                           CHGIII
  732 CONTINUE                                                           CHGIII
      WRITE(6,712)                                                       CHGIII
  712 FORMAT(12X,'NXLAB',15X,'XLAB')                                     CHGIII
      DO 733 IJ=1,NPLT                                                   CHGIII
```

```
      WRITE(6,713) IJ,NXLAB(IJ),(XLAB(L,IJ),L=1,NW60)                    CHGIII
  713 FORMAT(5X,'I.5.',I2,2X,I3,5X,15A4)                                 CHGIII
  733 CONTINUE                                                           CHGIII
      WRITE(6,714)                                                       CHGIII
  714 FORMAT(12X,'NYLAB',15X,'YLAB')                                     CHGIII
      DO 734 IJ=1,NPLT                                                   CHGIII
      WRITE(6,715) IJ,NYLAB(IJ),(YLAB(L,IJ),L=1,NW60)                    CHGIII
  715 FORMAT(5X,'I.6.',I2,2X,I3,5X,15A4)                                 CHGIII
  734 CONTINUE                                                           CHGIII
      WRITE(6,716)                                                       CHGIII
  716 FORMAT(12X,'NPLB1',15X,'PLB1')                                     CHGIII
      DO 735 IJ=1,NPLT                                                   CHGIII
      WRITE(6,717) IJ,NPLB1(IJ),(PLB1(L,IJ),L=1,NW60)                    CHGIII
  717 FORMAT(5X,'I.7.',I2,2X,I3,5X,15A4)                                 CHGIII
  735 CONTINUE                                                           CHGIII
      WRITE(6,718)                                                       CHGIII
  718 FORMAT(12X,'NPLB2',15X,'PLB2')                                     CHGIII
      DO 736 IJ=1,NPLT                                                   CHGIII
      WRITE(6,719) IJ,NPLB2(IJ),(PLB2(L,IJ),L=1,NW60)                    CHGIII
  719 FORMAT(5X,'I.8.',I2,2X,I3,5X,15A4)                                 CHGIII
  736 CONTINUE                                                           CHGIII
C                                                                        POSTPR
C     READ TIME HISTORY DATA FROM TAPE 8.                                POSTPR
C                                                                        POSTPR
   20 NPTS = 0                                                           POSTPR
      LINES = 0                                                          POSTPR
      IF  (NPRT(4).GT.0)  REWIND 8                                       POSTPR
      READ  (8,END=29)  NSEG,NJNT,NPL,NBLT,NBAG,NVEH,NGRND,NPANEL,       POSTPR
     *                  MNPL,MNBLT,MNSEG,MNBAG,MPL,MBLT,MSEG,MBAG        POSTPR
      READ  (8,END=29)  DATE,COMENT,VPSTTL,BDYTTL,BLTTTL,PLTTL,BAGTTL,   POSTPR
     *                  SEG,JOINT,UNITL,UNITM,UNITT,NSG,MSG,XSG,MCG,     ATBIII
     *                  MCGIN,KREF,NHRNSS,NBLTPH,NPTSPB,NSD,MSDM,MSDN    CHGIII
   21 READ  (8,END=29)  NT, UMSEC, ((TDATA(I,J),I=1,14),J=1,NT)          POSTPR
      R30 = 1.0D0                                                        TGMOD1
      IF(NPRT(30).GT.0) R30 = NPRT(30)                                   TGMOD1
      VDT1 = R30*PRDT                                                    TGMOD1
      TEST1 = DMOD(UMSEC,VDT1)                                           TGMOD1
      TEST1 = DMIN1(TEST1,DABS(VDT1 -TEST1))                             TGMOD1
      IF(NPRT(30).GT.0.AND.TEST1.GT.EPS(4)) GO TO 25                     TGMOD1
      NPTS = NPTS + 1                                                    POSTPR
      IF (NPTS.GT.NZD1 .AND. (NDPT.NE.0 .OR. LPLOT) ) STOP 52            ATBIII
      ZZ(NPTS,1) = UMSEC                                                 PLTINC
      Z(NPTS,1) = UMSEC                                                  PLTINC
      IF  (NDPT.EQ.0)  GO TO 22                                          POSTPR
C                                                                        POSTPR
C     STORE DATA FOR HIC, HSI AND CSI.                                   POSTPR
C                                                                        POSTPR
      JJ = 1                                                             POSTPR
      DO  61  I=1,2                                                      POSTPR
      IF  (JDTPTS(I).EQ.0)  GO TO 61                                     POSTPR
```

```
      JJ = JJ +1                                                        POSTPR
      JD = JDTPTS(I) - 1                                                POSTPR
      JE = 4*MOD(JD,3) + 4                                              POSTPR
      JP = JD/3 + 1                                                     POSTPR
      ZZ(NPTS,JJ) = TDATA(JE,JP)                                        PLTINC
   61 CONTINUE                                                          POSTPR
   22 IF  (.NOT.LPLOT)  GO TO 25                                        POSTPR
C                                                                       POSTPR
C     STORE DATA FOR PLOTTING                                           POSTPR
C                                                                       POSTPR
      JY =  1                                                           PLTINC
      DO  24  K=1,NPLT                                                  POSTPR
      JE = IABS(MX(2,K))                                                POSTPR
      IF  (JE.EQ.0)  GO TO 23                                           POSTPR
      JY = JY + 1                                                       POSTPR
      IF (JY.GT.NZD2) STOP 53                                           ATBIII
      JP = MX(1,K) - 20                                                 POSTPR
      Z(NPTS,JY) = TDATA(JE,JP)                                         POSTPR
   23 NYPLT = NYP(K)                                                    POSTPR
      DO  24  J=1,NYPLT                                                 POSTPR
      JY = JY + 1                                                       POSTPR
      JP = MY(1,J,K) - 20                                               POSTPR
      IF (JY.GT.NZD2) STOP 54                                           ATBIII
      JE = IABS(MY(2,J,K))                                              POSTPR
      Z(NPTS,JY) = UMSEC                                                POSTPR
   24 IF  (JE.NE.0)  Z(NPTS,JY) = TDATA(JE,JP)                          POSTPR
   25 IF  (.NOT.LTABH)  GO TO 21                                        POSTPR
C                                                                       POSTPR
C     STORE DATA TO PRINT TABULAR TIME HISTORIES                        POSTPR
C                                                                       POSTPR
      R26 = 1.0D0                                                       TGMOD1
      IF(NPRT(26).LT.0) IFLG = 1                                        TGMOD1
      IF(IFLG.EQ.1) N26 = IABS(NPRT(26))                                TGMOD1
      IF(IFLG.EQ.1) R26 = N26                                           TGMOD1
      VDT2 = R26*PRDT                                                   TGMOD1
      TEST2 = DMOD(UMSEC,VDT2)                                          TGMOD1
      TEST2 = DMIN1(TEST2,DABS(VDT2 - TEST2))                           TGMOD1
      IF  (NPRT(26).LE.0 .AND. TEST2.GT.EPS(4))  GO TO 21               TGMOD1
      LINES = LINES + 1                                                 POSTPR
      NTTH = MOD(LINES-1,LPP) + 1                                       POSTPR
      USEC(NTTH) = UMSEC                                                POSTPR
      DO  26  J=1,NT                                                    POSTPR
      DO  26  I=1,14                                                    POSTPR
   26 ZTTH(I,NTTH,J) = TDATA(I,J)                                       POSTPR
      IF  (NTTH.EQ.LPP)  CALL HEDING (LINES,LPP)                        POSTPR
      GO TO 21                                                          POSTPR
   29 IF  (.NOT.LTABH .OR. LINES.EQ.0)  GO TO 30                        POSTPR
      IF  (NTTH.NE.LPP)  CALL HEDING (LINES,LPP)                        POSTPR
   30 IF  (NDPT.NE.0)  CALL HICCSI(NPTS)                                POSTPR
      IF  (.NOT.LPLOT)  GO TO 98                                        POSTPR
```

```
C                                                                       POSTPR
C     PLOT DATA VIA SUBROUTINE SLPLOT.                                  POSTPR
C                                                                       POSTPR
C     INCLUDE ANY PROGRAM STATEMENTS HERE REQUIRED BY YOUR COMPUTER AND POSTPR
C     PLOTTING SYSTEMS FOR PLOT INITIALIZATION (E.G., CALL PLOTS).      POSTPR
C                                                                       POSTPR
      CALL PLOTS (0.0,0.0,10)                                           FIXSPT
      JZ = 1                                                            PLTINC
      DO  50  K=1,NPLT                                                  POSTPR
      JX = 1                                                            POSTPR
      IF  (MX(2,K).EQ.0)  GO TO 42                                      POSTPR
      JZ = JZ + 1                                                       POSTPR
      JX = JZ                                                           POSTPR
      IF  (Z(1,JX).EQ.0.0 .OR. MX(2,K).GE.0)  GO TO 42                  POSTPR
      DO  41  I=2,NPTS                                                  POSTPR
   41 Z(I,JX) = Z(I,JX) - Z(1,JX)                                       POSTPR
      Z(1,JX) = 0.0                                                     POSTPR
   42 NYPLT = NYP(K)                                                    POSTPR
      DO  44  J=1,NYPLT                                                 POSTPR
      JY = JZ + J                                                       POSTPR
      IF  (Z(1,JY).EQ.0.0 .OR. MY(2,J,K).GE.0)  GO TO 44                POSTPR
      DO  43  I=2,NPTS                                                  POSTPR
   43 Z(I,JY) = Z(I,JY) - Z(1,JY)                                       POSTPR
      Z(1,JY) = 0.0                                                     POSTPR
   44 CONTINUE                                                          POSTPR
      NXK = NX(K)                                                       POSTPR
      NYK = NY(K)                                                       POSTPR
      XOK = XO(K)                                                       POSTPR
      YOK = YO(K)                                                       POSTPR
      XNK = XN(K)                                                       POSTPR
      YNK = YN(K)                                                       POSTPR
      XLK = XL(K)                                                       POSTPR
      YLK = YL(K)                                                       POSTPR
      XSK = XS(K)                                                       POSTPR
      YSK = YS(K)                                                       POSTPR
      NXLABK = NXLAB(K)                                                 POSTPR
      NYLABK = NYLAB(K)                                                 POSTPR
      NPLB1K = NPLB1(K)                                                 POSTPR
      NPLB2K = NPLB2(K)                                                 POSTPR
      CALL SLPLOT(Z(1,JX  ), NXK, XOK, XNK, XLK, XSK, XLAB(1,K), NXLABK,POSTPR
     *            Z(1,JZ+1), NYK, YOK, YNK, YLK, YSK, YLAB(1,K), NYLABK,POSTPR
     *             NPTS,NYPLT,NZD1,PLB1(1,K),NPLB1K,PLB2(1,K),NPLB2K)   POSTPR
C                                                                       POSTPR
C     INSERT ANY CODE REQUIRED BY YOUR SYSTEM TO ADVANCE PLOT PAGES HEREPOSTPR
C                                                                       POSTPR
      IF(NPRT(31).EQ.1) GO TO 444                                       CHGIII
      X00 = -0.5*(XSK-(XLK-0.5)) + XLK + 3.0                            FXPLOT
      Y00 = -0.5*(YSK-(YLK-1.0))                                        FXPLOT
      CALL PLOT (X00,Y00,-3)                                            FXPLOT
   50 JZ = JZ + NYPLT                                                   POSTPR
```

```
  444 CONTINUE                                                       CHGIII
C                                                                    POSTPR
C     INSERT ANY PLOT TERMINATION CODE REQUIRED BY YOUR SYSTEM HERE. POSTPR
C                                                                    POSTPR
      CALL PLOT(12.0,0.0,999)                                        PECONV
   98 CALL ELTIME (2,36)                                             POSTPR
   99 RETURN                                                         POSTPR
      END                                                            POSTPR
```

```
      SUBROUTINE PRINT(SUB)                                             PRINT
C                                                  REV IV    07/24/86SLIP
C     SUBROUTINE TO PRINT SEGMENT LINEAR AND ANGULAR                    PRINT
C     POSITIONS, VELOCITIES AND ACCELERATIONS FOR A GIVEN TIME.         PRINT
C                                                                       PRINT
C     ARGUMENTS                                                         PRINT
C        SUB: CALLING SUBROUTINE NAME                                   PRINT
C                                                                       PRINT
      IMPLICIT REAL*8(A-H,O-Z)                                          PRINT
      COMMON/CONTRL/ TIME,NSEG,NJNT,NPL,NBLT,NBAG,NVEH,NGRND,           PRINT
     *               NS,NQ,NSD,NFLX,NHRNSS,NWINDF,NJNTF,NPRT(36),NPG    PAGE
      COMMON/SGMNTS/ D(3,3,30),WMEG(3,30),WMEGD(3,30),U1(3,30),U2(3,30),PRINT
     *               SEGLP(3,30),SEGLV(3,30),SEGLA(3,30),NSYM(30)       PRINT
      COMMON/DESCRP/ PHI(3,30),W(30),RW(30),SR(4,60),HA(3,60),HB(3,60), SLIP
     *               RPHI(3,30),HT(3,3,60),SPRING(5,90),VISC(7,90),     PRINT
     *               JNT(30),IPIN(30),ISING(30),IGLOB(30),JOINTF(30)    PRINT
      COMMON/CMATRX/ V1(3,30),V2(3,30),V3(3,12),B12(3,3,60),A22(3,3,60),PRINT
     *               F(3,30),TQ(3,30),WJ(30),A11(3,3,30)                SLIP
      COMMON/CSTRNT/ A13(3,3,24),A23(3,3,24),B31(3,3,24),B32(3,3,24),   PRINT
     *               HHT(3,3,12),RK1(3,12),RK2(3,12),QQ(3,12),TQQ(3,12),PRINT
     *               RQQ(3,12),HQQ(3,12),SQQ(12),CFQQ(12),              PRINT
     *               KQ1(12),KQ2(12),KQTYPE(12)                         PRINT
      COMMON/TITLES/ DATE(3),COMENT(40),VPSTTL(20),BDYTTL(5),           PRINT
     *               BLTTTL(5,8),PLTTL(5,30),BAGTTL(5,6),SEG(30),       PRINT
     *               JOINT(30),CGS(30),JS(30)                           PRINT
      REAL DATE,COMENT,VPSTTL BDYTTL,BLTTTL,PLTTL,BAGTTL,SEG,JOINT      PRINT
      LOGICAL*1 CGS,JS                                                  PRINT
      COMMON/CEULER/ IEULER(30),HIR(3,3,90),ANG(3,30),ANGD(3,30),       JDRIFT
     *               FE(3,30),TQE(3,30),CONST(5,30)                     JDRIFT
      COMMON/CNSNTS/ PI,RADIAN,G,THIRD,EPS(24),                         PRINT
     *               UNITL,UNITM,UNITT,GRAVTY(3),TWOPI                  TWOPI
      COMMON/RSAVE/  XSG(3,20,3),DPMI(3,3,30),LPMI(30),                 ATBIII
     *               NSG(9),MSG(20,9),MCG,MCGIN(24,5),KREF(20,9)        TTHKREF
      COMMON/TEMPVS/ YPR(3),T1(3),T2(3),HH(3),T3(3,3),SKE(3),TKE(3),V   KINETIC
C                                                                       PRINT
      IPC = 1                                                           PRINT
      TMSEC = 1000.0*TIME                                               PRINT
      WRITE (6,11) IPC,SUB,TMSEC,NPG                                    PAGE
      NPG=NPG+1                                                         PAGE
   11 FORMAT (I1,6X,A6,' FUNCTIONS FOR TIME=',                          CHGIII
     *        F10.3,' MSEC',75X,'PAGE',I5/)                             PAGE
      WRITE (6,760)                                                     CHGIII
  760 FORMAT(1X,23X,'(INERTIAL)',29X,'(LOCAL)',35X,'(LOCAL)')           CHGIII
      WRITE (6,12) UNITT,UNITT                                          PRINT
   12 FORMAT(19X,'ANGULAR ROTATION (DEG)',                              PRINT
     *       12X,'ANGULAR VELOCITY (RAD/',A4,')',                       PRINT
     *       12X,'ANGULAR ACCELERATION (RAD/',A4,'**2)'/                PRINT
     *       ' SEGMENT',9X,'YAW' ,7X,'PITCH',7X,'ROLL',                 PRINT
     *       11X,'X',11X,'Y',11X,'Z',15X,'X',13X,'Y',13X,'Z'/)          PRINT
      MBAG = NVEH + NBAG                                                PRINT
```

```
      DO  20  I=1,MBAG                                                  PRINT
      IF  (LPMI(I).EQ.0)  GO TO 19                                      PRINT
      CALL DOT33 (DPMI(1,1,I),D(1,1,I),T3)                              PRINT
      CALL DOT31(DPMI(1,1,I),WMEG(1,I),T1)                              FIXPRT
      CALL DOT31(DPMI(1,1,I),WMEGD(1,I),T2)                             FIXPRT
      CALL YPRDEG (T3,YPR)                                              PRINT
      WRITE (6,31) I,SEG(I),YPR,(T1(K),K=1,3),(T2(K),K=1,3)             FIXPRT
      GO TO 20                                                          PRINT
   19 CALL YPRDEG (D(1,1,I),YPR)                                        PRINT
      WRITE  (6,31)  I,SEG(I),YPR,(WMEG(K,I),K=1,3),(WMEGD(K,I),K=1,3)  FIXPRT
   20 CONTINUE                                                          FIXPRT
      WRITE(6,770)                                                      CHGIII
  770 FORMAT(//,1X,23X,'(INERTIAL)',27X,'(INERTIAL)',32X,'(INERTIAL)')  CHGIII
      WRITE  (6,22)  UNITL,UNITL,UNITT                                  PRINT
   22 FORMAT(18X,'LINEAR POSITION (',A4,')',                            CHGIII
     *          13X,'LINEAR VELOCITY (',A4,'/',A4,')',                  PRINT
     *          16X,'LINEAR ACCELERATIONS (G''S)'/                      PRINT
     *       ' SEGMENT',10X,'X',10X,'Y',10X,'Z',                        PRINT
     *       13X,'X',11X,'Y',11X,'Z',15X,'X',13X,'Y',13X,'Z'/)          PRINT
      DO  30  I=1,MBAG                                                  PRINT
      DO  29  K=1,3                                                     PRINT
   29 T1(K) = SEGLA(K,I)/G                                              PRINT
   30 WRITE (6,31) I,SEG(I),(SEGLP(K,I),K=1,3),(SEGLV(K,I),K=1,3),T1    PRINT
   31 FORMAT(I3,1X,A4,3X,3F11.4,3X,3F12.5,3X,3F14.6)                    PRINT
      IF (NSEG.GT.6) WRITE (6,32) NPG                                   PAGE
      IF (NSEG.GT.6) NPG=NPG+1                                          PAGE
   32 FORMAT('1',122X,'PAGE',I5)                                        PAGE
      WRITE(6,775)                                                      CHGIII
  775 FORMAT(//,1X,23X,'(INERTIAL)',29X,'(LOCAL)')                      CHGIII
      WRITE (6,33) UNITL,UNITT,UNITT,UNITM,UNITL                        KINETIC
   33 FORMAT(18X,'U1 ARRAY (',A4,'/',A4,'**2)',                         KINETIC
     *          14X,'U2 ARRAY (RAD/',A4,'**2)',                         KINETIC
     *          25X,'KINETIC ENERGY'/                                   KINETIC
     *          15X,'EXTERNAL LINEAR ACCELERATIONS',                    KINETIC
     *           8X,'EXTERNAL ANGULAR ACCELERATIONS',                   KINETIC
     *          22X,'(',A4,'-',A4,')'/                                  KINETIC
     * ' SEGMENT',10X,'X',10X,'Y',10X,'Z',13X,'X',11X,'Y',11X,'Z',      KINETIC
     *           14X,'LINEAR',7X,'ANGULAR',7X,'TOTAL'/)                 KINETIC
      DO 80 J=1,3                                                       KINETIC
80    TKE(J)=0.0                                                        KINETIC
      DO 34 I=1,NSEG                                                    PRINT
      V=SEGLV(1,I)**2+SEGLV(2,I)**2+SEGLV(3,I)**2                       KINETIC
      SKE(1)=0.5*W(I)*V/G                                               KINETIC
      SKE(2)=0.0                                                        KINETIC
      DO 81 J=1,3                                                       KINETIC
81    SKE(2)=SKE(2)+0.5*PHI(J,I)*WMEG(J,I)**2                           KINETIC
      SKE(3)=SKE(1)+SKE(2)                                              KINETIC
      DO 82 J=1,3                                                       KINETIC
82    TKE(J)=TKE(J)+SKE(J)                                              KINETIC
      IF  (LPMI(I).EQ.0) GO TO 73                                       FIXPRT
```

```
      CALL DOT31 (DPMI(1,1,I),U2(1,I),T1)                                 FIXPRT
      WRITE (6,61) I,SEG(I),(U1(K,I),K=1,3),                              KINETIC
     *             (T1(K),K=1,3),(SKE(K),K=1,3)                           KINETIC
      GO TO 34                                                            PRINT
   73 CONTINUE                                                            PRINT
      WRITE (6,61) I,SEG(I),(U1(K,I),K=1,3),                              KINETIC
     *             (U2(K,I),K=1,3),(SKE(K),K=1,3)                         KINETIC
   61 FORMAT(I3,1X,A4,3X,3(D11.4,1X),3X,3(D12.5,1X),3X,3(D12.5,1X))       KINETIC
   34 CONTINUE                                                            FIXPRT
      WRITE(6,83) (TKE(K),K=1,3)                                          KINETIC
83    FORMAT(1X,98X,'TOTAL BODY KINETIC ENERGY'/                          KINETIC
     *       1X,90X,3(1X,D12.5))                                          KINETIC
      IF (NJNT.LE.0) GO TO 39                                             PRINT
      WRITE(6,776)                                                        CHGIII
  776 FORMAT(//,1X,27X,'(INERTIAL)',27X,'(INERTIAL)')                     CHGIII
      WRITE (6,35) UNITM,UNITL,UNITM,UNITT                                PRINT
   35 FORMAT(24X,'JOINT FORCES (',A4,')',                                 CHGIII
     *          15X,'JOINT TORQUES (',2A4,')',                            PRINT
     *           9X,'RELATIVE ANGULAR'/                                   PRINT
     *'  JOINT  IPIN',9X,'X',10X,'Y',10X,'Z',13X,'X',11X,'Y',11X,'Z',     PRINT
     *           7X,'VELOCITY (RAD/',A4,')'/)                             PRINT
      DO 36 J=1,NJNT                                                      PRINT
      IPINJ = IPIN(J)                                                     PRINT
      IF (IABS(IPIN(J)).EQ.4) IPINJ = IEULER(J)                           PRINT
      DO 137 II=1,3                                                       MISC
137   T1(II)=-TQ(II,J)                                                    MISC
      WRITE (6,37) J,JOINT(J),IPINJ,(F(K,J),K=1,3),(T1(K),K=1,3),WJ(J)    MISC
   37 FORMAT(I3,1X,A4,I4,7X,3(D10.3,1X),3X,3(D11.4,1X),3X,F13.3)          FIXPRT
   36 CONTINUE                                                            FIXPRT
   39 IF (NQ.LE.0) GO TO 99                                               PRINT
      WRITE (6,41)                                                        CHGIII
      WRITE (6,47)                                                        CHGIII
   47 FORMAT(1X,45X,'(INERTIAL)')                                         CHGIII
      WRITE (6,49) UNITM,UNITL                                            CHGIII
   41 FORMAT(///' OTHER CONSTRAINT FORCES',/)                             CHGIII
   49 FORMAT(1X,' NO.  TYPE   SEG1  SEG2',                                CHGIII
     *       15X,'CONSTRAINT FORCE (',A4,')',                             PRINT
     *       16X,'DISTANCE (',A4,')'/)                                    PRINT
      ICH = 0                                                             FIXPRT
      DO 50 J=1,NQ                                                        PRINT
      IF (KQTYPE(J).NE.5) ICH = 0                                         FIXPRT
      IF (KQTYPE(J).LT.0) GO TO 50                                        PRINT
      IF (KQTYPE(J).EQ.5) ICH = ICH + 1                                   FIXPRT
      IF (ICH.EQ.2) GO TO 50                                              FIXPRT
      M = KQ1(J)                                                          PRINT
      N = KQ2(J)                                                          PRINT
      CALL DOT31(D(1,1,M),RK1(1,J),T1)                                    PRINT
      CALL DOT31(D(1,1,N),RK2(1,J),T2)                                    PRINT
      S1 = 0.0                                                            PRINT
      DO 42 I=1,3                                                         PRINT
```

```
   HH(I) = SEGLP(I,M)+T1(I) - SEGLP(I,N)-T2(I)                           PRINT
42 S1 = S1 + HH(I)**2                                                    PRINT
   SQS1 = DSQRT(S1)                                                      PRINT
   WRITE (6,43) J,KQTYPE(J),SEG(M),SEG(N),(QQ(I,J),I=1,3),SQS1           PRINT
43 FORMAT(I4,I6,4X,A4,2X,A4,3X,3G15.7,6X,G15.7)                          PRINT
50 CONTINUE                                                              PRINT
99 IF  (NPRT(28).LE.0)  NPRT(28) = -1                                    PRINT
   RETURN                                                                PRINT
   END                                                                   PRINT
```

```
      SUBROUTINE PRIPLT                                                 PRIPLT
C                                              REV IV    07/24/86SLIP
C     PRODUCES PRINTER PLOT OF Y-Z PLANE VIEW AND X-Z PLANE VIEW OF     PRIPLT
C     BODY SEGMENT CGS, JOINTS AND SELECTED POINTS OF VEHICLE COMPONENTSPRIPLT
C                                                                       PRIPLT
      IMPLICIT REAL*8 (A-H,O-Z)                                         PRIPLT
      COMMON/CONTRL/ TIME,NSEG,NJNT,NPL,NBLT,NBAG,NVEH,NGRND,           PRIPLT
     *               NS,NQ,NSD,NFLX,NHRNSS,NWINDF,NJNTF,NPRT(36),NPG    PAGE
      COMMON/SGMNTS/ D(3,3,30),WMEG(3,30),WMEGD(3,30),U1(3,30),U2(3,30),PRIPLT
     *               SEGLP(3,30),SEGLV(3,30),SEGLA(3,30),NSYM(30)       PRIPLT
      COMMON/DESCRP/ PHI(3,30),W(30),RW(30),SR(4,60),HA(3,60),HB(3,60), SLIP
     *               RPHI(3,30),HT(3,3,60),SPRING(5,90),VISC(7,90),     PRIPLT
     *               JNT(30),IPIN(30),ISING(30),IGLOB(30),JOINTF(30)    PRIPLT
      COMMON/JBARTZ/ MNPL(   30),MNBLT(   8),MNSEG(   30),MNBAG(    6), PRIPLT
     *               MPL(3,5,30),MBLT(3,5,8),MSEG(3,5,30),MBAG(3,10,6), PRIPLT
     *               NTPL( 5,30),NTBLT( 5,8),NTSEG( 5,30)               PRIPLT
      COMMON/CNTSRF/ PL(24,30),BELT(20,8),TPTS(6,8),BD(24,40)           EDGE
      COMMON/TITLES/ DATE(3),COMENT(40),VPSTTL(20),BDYTTL(5),           PRIPLT
     *               BLTTTL(5,8),PLTTL(5,30),BAGTTL(5,6),SEG(30),       PRIPLT
     *               JOINT(30),CGS(30),JS(30)                           PRIPLT
      COMMON/HRNESS/ BAR(15,100),BB(100),BBDOT(100),PLOSS(2,100),       PRIPLT
     *               XLONG(20),HTIME(2),IBAR(5,100),NL(2,100),          PRIPLT
     *               NPTSPB(20),NPTPLY(20),NTHRNS(20),NBLTPH(5)         PRIPLT
      REAL DATE,COMENT,VPSTTL,BDYTTL,BLTTTL,PLTTL,BAGTTL,SEG,JOINT      PRIPLT
      LOGICAL*1 CGS,JS                                                  PRIPLT
      COMMON/TEMPVS/ TEMP1(3),TEMP(3),TEMP2(3),CJOINT(3,30),BSN(2),     PRIPLT
     *               PLOTYZ(96,55),PLOTXZ(96,55),PLOTXY(96,55)          PRIPLT
      LOGICAL*1 PLOTYZ,PLOTXZ,PLOTXY,CHARS(7),BLANK,BCHAR               PRIPLT
      LOGICAL   NPRT5,NPRT6,NPRT7                                       PRIPLT
      DATA CHARS/'.','+','@','!','_','-','*'/, BLANK/' '/               PRIPLT
      DATA ISTEP/0/ , NPLTI/96/ , NPLTJ/55/                             PRIPLT
C                                                                       PRIPLT
C     DETERMINE IF PLOTTING IS TO BE DONE FOR THIS TIME STEP.           PRIPLT
C                                                                       PRIPLT
      ISTEP = ISTEP+1                                                   PRIPLT
      NPRT5 = (NPRT(5).EQ.1)                                            PRIPLT
      IF (NPRT(5).GT.1) NPRT5 = (MOD(ISTEP,NPRT(5)).EQ.1)               PRIPLT
      NPRT6 = (NPRT(6).EQ.1)                                            PRIPLT
      IF (NPRT(6).GT.1) NPRT6 = (MOD(ISTEP,NPRT(6)).EQ.1)               PRIPLT
      NPRT7 = (NPRT(7).EQ.1)                                            PRIPLT
      IF (NPRT(7).GT.1) NPRT7 = (MOD(ISTEP,NPRT(7)).EQ.1)               PRIPLT
      IF  (.NOT.NPRT5 .AND. .NOT.NPRT6 .AND. .NOT.NPRT7)  GO TO 99      PRIPLT
      CALL ELTIME(1, 4)                                                 PRIPLT
C                                                                       PRIPLT
C     BLANK OUT PLOT ARRAYS.                                            PRIPLT
C                                                                       PRIPLT
      DO  10  J=1,NPLTJ                                                 PRIPLT
      PLOTYZ(1,J) = CHARS(6)                                            PRIPLT
      PLOTXZ(1,J) = CHARS(6)                                            PRIPLT
      PLOTXY(1,J) = CHARS(6)                                            PRIPLT
```

```
      DO  10  I=2,NPLTI                                                 PRIPLT
      PLOTYZ(I,J) = BLANK                                               PRIPLT
      PLOTXZ(I,J) = BLANK                                               PRIPLT
   10 PLOTXY(I,J) = BLANK                                               PRIPLT
C                                                                       PRIPLT
C     PLOT VEHICLE REFERENCE ORIGIN USING SYMBOL(*).                    PRIPLT
C                                                                       PRIPLT
      CALL PLTXYZ (SEGLP(1,NVEH),CHARS(7))                              PRIPLT
C                                                                       PRIPLT
C     PLOT CG OF BODY SEGMENTS USING SEGMENT SYMBOLS.                   PRIPLT
C                                                                       PRIPLT
      DO 20  I=1,NSEG                                                   PRIPLT
   20 CALL PLTXYZ(SEGLP(1,I),CGS(I))                                    PRIPLT
C                                                                       PRIPLT
C     COMPUTE AND PLOT JOINT LOCATIONS USING JOINT SYMBOLS.             PRIPLT
C                                                                       PRIPLT
      IF  (NJNT.EQ.0)  GO TO 40                                         PRIPLT
      DO  31  J=1,NJNT                                                  PRIPLT
      I = IABS(JNT(J))                                                  PRIPLT
      IF  (I.LE.0)  GO TO 31                                            PRIPLT
      CALL DOT31(D(1,1,I),SR(1,2*J-1),TEMP)                             PRIPLT
      DO 30 L=1,3                                                       PRIPLT
   30 CJOINT(L,J) = TEMP(L)+SEGLP(L,I)                                  PRIPLT
      CALL PLTXYZ (CJOINT(1,J),JS(J))                                   PRIPLT
   31 CONTINUE                                                          PRIPLT
      IF  (NPRT(13).NE.0)  WRITE (6,32) ((CJOINT(I,J),I=1,3),J=1,NJNT)  PRIPLT
   32 FORMAT ('0 JOINT POSITIONS'/(1X,9F14.4))                          PRIPLT
C                                                                       PRIPLT
C     PLOT BELT ANCHOR, FIXED AND TANGENT POINTS USING SYMBOL(.).       PRIPLT
C                                                                       PRIPLT
   40 IF  (NBLT.LE.0)  GO TO 50                                         PRIPLT
      DO  43  J=1,NBLT                                                  PRIPLT
      IF  (MNBLT(J).LE.0)  GO TO 43                                     PRIPLT
      M1 = MBLT(1,1,J)                                                  PRIPLT
      M2 = MBLT(2,1,J)                                                  PRIPLT
      M3 = MBLT(3,1,J)                                                  PRIPLT
      DO  41  I=1,3                                                     PRIPLT
   41 TEMP1(I) = BELT(I+6,J) + BD(I+3,M3)                               PRIPLT
      CALL DOT31 (D(1,1,M2),TEMP1,TEMP)                                 PRIPLT
      CALL DOT31 (D(1,1,M1),BELT(1,J),TEMP1)                            PRIPLT
      CALL DOT31 (D(1,1,M1),BELT(4,J),TEMP2)                            PRIPLT
      DO  42  I=1,3                                                     PRIPLT
      TEMP1(I) = TEMP1(I) + SEGLP(I,M1)                                 PRIPLT
      TEMP2(I) = TEMP2(I) + SEGLP(I,M1)                                 PRIPLT
   42 TEMP (I) = TEMP (I) + SEGLP(I,M2)                                 PRIPLT
      CALL PLTXYZ (TEMP1     ,CHARS(1))                                 PRIPLT
      CALL PLTXYZ (TEMP2     ,CHARS(1))                                 PRIPLT
      CALL PLTXYZ (TEMP      ,CHARS(1))                                 PRIPLT
      CALL PLTXYZ (TPTS(1,J),CHARS(1))                                  PRIPLT
      CALL PLTXYZ (TPTS(4,J),CHARS(1))                                  PRIPLT
```

```
   43 CONTINUE                                                          PRIPLT
C                                                                       PRIPLT
C     PLOT POINTS IN PLAY ON HARNESS-BELT SYSTEMS USING SYMBOL(.).      PRIPLT
C                                                                       PRIPLT
   50 IF  (NHRNSS.LE.0)  GO TO 60                                       PRIPLT
      J1 = 1                                                            PRIPLT
      K1 = 1                                                            PRIPLT
      DO  54  NH=1,NHRNSS                                               PRIPLT
      IF  (NBLTPH(NH).LE.0)  GO TO 54                                   PRIPLT
      J2 = J1 + NBLTPH(NH) - 1                                          PRIPLT
      DO  53  NB=J1,J2                                                  PRIPLT
      IF  (NPTPLY(NB).LE.0)  GO TO 53                                   PRIPLT
      K2 = K1 + NPTPLY(NB) - 1                                          PRIPLT
      DO  52   K=K1,K2                                                  PRIPLT
      KI = NL(1,K)                                                      PRIPLT
      KS = IABS(IBAR(1,KI))                                             PRIPLT
      IF  (KS.GT.100)  KS = MOD(KS,100)                                 PRIPLT
      CALL DOT31 (D(1,1,KS),BAR(4,KI),TEMP1)                            PRIPLT
      CALL DOT31 (D(1,1,KS),BAR(7,KI),TEMP2)                            PRIPLT
      DO  51  I=1,3                                                     PRIPLT
   51 TEMP(I) = SEGLP(I,KS) + TEMP1(I) + TEMP2(I)                       PRIPLT
   52 CALL PLTXYZ (TEMP,CHARS(1))                                       PRIPLT
      K1 = K2+1                                                         PRIPLT
   53 CONTINUE                                                          PRIPLT
      J1 = J2+1                                                         PRIPLT
   54 CONTINUE                                                          PRIPLT
C                                                                       PRIPLT
C     PLOT CENTER AND END OF AXES OF ELLIPSOIDAL TARGET USING SYMBOLS   PRIPLT
C     (*) FOR CENTER, (-) FOR ENDS OF Z AXIS,(!) FOR ENDS OF X,Y AXES.  PRIPLT
C                                                                       PRIPLT
   60 IF  (NBAG.EQ.0)  GO TO 80                                         PRIPLT
      BSN(1) =  1.0                                                     PRIPLT
      BSN(2) = -1.0                                                     PRIPLT
      DO  68  J=1,NBAG                                                  PRIPLT
      IF  (MNBAG(J).EQ.0)  GO TO 68                                     PRIPLT
      JB = NVEH+J                                                       PRIPLT
      BCHAR = CHARS(5)                                                  PRIPLT
      L2 = 2                                                            PRIPLT
      DO  67  I=1,4                                                     PRIPLT
      IF  (I.EQ.3)  BCHAR = CHARS(4)                                    PRIPLT
      IF  (I.EQ.4)  BCHAR = CHARS(3)                                    PRIPLT
      IF  (I.EQ.4)  L2 = 1                                              PRIPLT
      DO  67  L=1,L2                                                    PRIPLT
      DO  64  K=1,3                                                     PRIPLT
   64 TEMP1(K) = BD(K+3,JB)                                             PRIPLT
      IF  (I.EQ.4)  GO TO 65                                            PRIPLT
      TEMP1(I) = TEMP1(I) + BSN(L)*BD(I,JB)                             PRIPLT
   65 CALL DOT31 (D(1,1,JB),TEMP1,TEMP2)                                PRIPLT
      DO  66  K=1,3                                                     PRIPLT
   66 TEMP2(K) = TEMP2(K) + SEGLP(K,JB)                                 PRIPLT
```

```
   67 CALL PLTXYZ (TEMP2,BCHAR)                                              PRIPLT
   68 CONTINUE                                                               PRIPLT
C                                                                            PRIPLT
C     PRINT Y-Z , X-Z AND X-Y PLANE VIEW PLOTS.                              PRIPLT
C                                                                            PRIPLT
   80 TMSC = 1000.0*TIME                                                     PRIPLT
      IF  (.NOT.NPRT5)  GO TO 83                                             PRIPLT
      WRITE  (2,81)  TMSC,SEGLP(2,NVEH),SEGLP(3,NVEH)                        PRIPLT
   81 FORMAT ('1 T=',F10.3,'   Y0=',F10.5,'   Z0=',F10.5,'   Y-Z PLANE')PRIPLT
      WRITE  (2,82)  PLOTYZ                                                  PRIPLT
   82 FORMAT (2X,96A1)                                                       PRIPLT
   83 IF  (.NOT.NPRT6)  GO TO 85                                             PRIPLT
      WRITE  (2,84)  TMSC,SEGLP(1,NVEH),SEGLP(3,NVEH)                        PRIPLT
   84 FORMAT ('1 T=',F10.3,'   X0=',F10.5,'   Z0=',F10.5,'   X-Z PLANE')PRIPLT
      WRITE  (2,82)  PLOTXZ                                                  PRIPLT
   85 IF  (.NOT.NPRT7)  GO TO 87                                             PRIPLT
      WRITE  (2,86)  TMSC,SEGLP(1,NVEH),SEGLP(2,NVEH)                        PRIPLT
   86 FORMAT ('1 T=',F10.3,'   X0=',F10.5,'   Y0=',F10.5,'   X-Y PLANE')PRIPLT
      WRITE  (2,82)  PLOTXY                                                  PRIPLT
   87 CALL ELTIME(2, 4)                                                      PRIPLT
   99 RETURN                                                                 PRIPLT
      END                                                                    PRIPLT
```

```
      SUBROUTINE QSET(F,Y,X,DER,N)                                      QSET
C                                                   REV III.3 10/01/84REVIII
      IMPLICIT REAL*8(A-H,O-Z)                                          QSET
      DIMENSION F(5,3,80),Y(5,3,80),X(3,80),DER(3,80)                   QSET
      DIMENSION T1(3),T2(3),T3(3),T4(3)                                 QSET
      DO  20  I=1,N                                                     QSET
      E1=DSQRT(1.D0 -X(1,I)**2-X(2,I)**2-X(3,I)**2)                     QSET
      E1D=-(X(1,I)*DER(1,I)+X(2,I)*DER(2,I)+X(3,I)*DER(3,I))/E1         QSET
      E2=DSQRT(1.D0-Y(1,1,I)**2-Y(1,2,I)**2-Y(1,3,I)**2)                QSET
      E2D=-(Y(1,1,I)*Y(2,1,I)+Y(1,2,I)*Y(2,2,I)+Y(1,3,I)*Y(2,3,I))/E2   QSET
      UHB=X(1,I)*F(3,1,I)+X(2,I)*F(3,2,I)+X(3,I)*F(3,3,I)               QSET
      UHC=X(1,I)*F(4,1,I)+X(2,I)*F(4,2,I)+X(3,I)*F(4,3,I)               QSET
      UDB=DER(1,I)*F(3,1,I)+DER(2,I)*F(3,2,I)+DER(3,I)*F(3,3,I)         QSET
      UDD=DER(1,I)**2+DER(2,I)**2+DER(3,I)**2                           QSET
      EB=(E1D**2+UDD+UHB)/E1                                            QSET
      EC =  (1.5*(UDB-E1D*EB)+UHC+F(5,1,I)*(E1D**2+UDD))/E1             QSET
      T1(1)=X(2,I)*F(3,3,I)-X(3,I)*F(3,2,I)                             QSET
      T2(1)=X(2,I)*F(4,3,I)-X(3,I)*F(4,2,I)                             QSET
      T3(1)=X(2,I)*Y(1,3,I)-X(3,I)*Y(1,2,I)                             QSET
      T4(1)=X(2,I)*Y(2,3,I)-X(3,I)*Y(2,2,I)                             QSET
      T1(2)=X(3,I)*F(3,1,I)-X(1,I)*F(3,3,I)                             QSET
      T2(2)=X(3,I)*F(4,1,I)-X(1,I)*F(4,3,I)                             QSET
      T3(2)=X(3,I)*Y(1,1,I)-X(1,I)*Y(1,3,I)                             QSET
      T4(2)=X(3,I)*Y(2,1,I)-X(1,I)*Y(2,3,I)                             QSET
      T1(3)=X(1,I)*F(3,2,I)-X(2,I)*F(3,1,I)                             QSET
      T2(3)=X(1,I)*F(4,2,I)-X(2,I)*F(4,1,I)                             QSET
      T3(3)=X(1,I)*Y(1,2,I)-X(2,I)*Y(1,1,I)                             QSET
      T4(3)=X(1,I)*Y(2,2,I)-X(2,I)*Y(2,1,I)                             QSET
      DO 20 J=1,3                                                       QSET
      F(3,J,I)=E1*F(3,J,I)-T1(J)+EB*X(J,I)                              JTF984
      F(4,J,I)=E1*F(4,J,I)-T2(J)+EC*X(J,I)                              JTF984
      Y(3,J,I)=E1*Y(1,J,I)-T3(J)-E2*X(J,I)                              JTF984
   20 Y(4,J,I)=E1*Y(2,J,I)-T4(J)-E2D*X(J,I)                             JTF984
      RETURN                                                            QSET
      END                                                               QSET
```

```
      SUBROUTINE QUAT(ANG,Q)                                             QUAT
C                                                   REV IV    07/23/86TWOPI
C     COMPUTES QUATERNIONS FROM YAW, PITCH, ROLL ANGLES IN DEGREES      QUAT
      IMPLICIT REAL *8(A-H,O-Z)                                          QUAT
      DIMENSION ANG(3),Q(4),R(4),T(3)                                    QUAT
      COMMON/CNSNTS/ PI,RADIAN,G,THIRD,EPS(24),                          QUAT
     *               UNITL,UNITM,UNITT,GRAVTY(3),TWOPI                   TWOPI
      A = 0.5*ANG(1)*RADIAN                                              QUAT
      Q(1) = DCOS(A)                                                     QUAT
      Q(2) = 0.0                                                         QUAT
      Q(3) = 0.0                                                         QUAT
      Q(4) = DSIN(A)                                                     QUAT
      K = 3                                                              QUAT
      DO 10 I = 2,3                                                      QUAT
      A = 0.5*ANG(I)*RADIAN                                              QUAT
      R(1) = DCOS(A)                                                     QUAT
      R(2) = 0.0                                                         QUAT
      R(3) = 0.0                                                         QUAT
      R(4) = 0.0                                                         QUAT
      R(K) = DSIN(A)                                                     QUAT
      DOT = Q(2)*R(2) + Q(3)*R(3) + Q(4)*R(4)                            QUAT
      CALL CROSS(Q(2),R(2),T)                                            QUAT
      DO 5 J = 2,4                                                       QUAT
    5 Q(J) = Q(1)*R(J) + R(1)*Q(J) + T(J-1)                              QUAT
      Q(1) = Q(1)*R(1) - DOT                                             QUAT
   10 K = 2                                                              QUAT
      SUM = DSQRT(Q(1)**2 + Q(2)**2 + Q(3)**2 + Q(4)**2)                 QUAT
      DO 12 I = 1,4                                                      QUAT
   12  Q(I) = Q(I)/SUM                                                   QUAT
       RETURN                                                            QUAT
      END                                                                QUAT
```

```
      DOUBLE PRECISION FUNCTION RCRT(A,PL,Z,IP)                         RCRT
C                                                  REV 03    07/19/73RCRT
C     COMPUTES THE RADIUS OF CURVATURE AT POINT Z OF ELLIPSOID A        RCRT
C     IN THE PLANE PL(I,IP) WHERE                                       RCRT
C                                                                       RCRT
C          A: 3X3 MATRIX DEFINING ELLIPSOID.                            RCRT
C         PL: 4X3 MATRIX CONTAINING THREE ORTHONORMAL VECTORS.          RCRT
C          Z: 3 COORDINATES OF POINT ON THE ELLIPSOID                   RCRT
C             AS MEASURED FROM CENTER OF ELLIPSOID                      RCRT
C         IP: IDENTIFIES THE NORMAL VECTOR OF PLANE IN WHICH THE        RCRT
C             RADIUS OF CURVATURE IS DESIRED.                           RCRT
C                                                                       RCRT
      IMPLICIT REAL*8 (A-H,O-Z)                                         RCRT
      DIMENSION A(3,3),PL(4,3),Z(3),T(5)                                RCRT
      DO 10  I=1,5                                                      RCRT
   10 T(I) = 0.0                                                        RCRT
      M = IP+1                                                          RCRT
      N = IP+2                                                          RCRT
      IF(M.GT.3) M = M-3                                                RCRT
      IF(N.GT.3) N = N-3                                                RCRT
      DO 30  I=1,3                                                      RCRT
      S1 = 0.                                                           RCRT
      S2 = 0.                                                           RCRT
      DO 20  J=1,3                                                      RCRT
      S1 = S1+A(I,J)*PL(J,M)                                            RCRT
   20 S2 = S2+A(I,J)*PL(J,N)                                            RCRT
      T(1) = T(1)+S1*Z(I)                                               RCRT
      T(2) = T(2)+S2*Z(I)                                               RCRT
      T(3) = T(3)+S1*PL(I,M)                                            RCRT
      T(4) = T(4)+S2*PL(I,N)                                            RCRT
   30 T(5) = T(5)+S1*PL(I,N)                                            RCRT
      W = DSQRT(T(1)**2+T(2)**2)                                        RCRT
      T(1) = T(1)/W                                                     RCRT
      T(2) = T(2)/W                                                     RCRT
      RCRT = W/(T(3)*T(2)**2-2.0*T(1)*T(2)*T(5)+T(4)*T(1)**2)           RCRT
      IF(RCRT.LT.0.0) RCRT = -RCRT                                      RCRT
      RETURN                                                            RCRT
      END                                                               RCRT
```

```
      SUBROUTINE ROT (A,L,TH)                                           ROT
C                                                    REV IV    07/23/86TWOPI
C     COMPUTES ROTATION MATRIX A FOR ANGLE TH                           ROT
C     ABOUT X,Y OR Z AXIS AS L = 1,2, OR 3.                             ROT
C                                                                       ROT
C       ARGUMENTS:                                                      ROT
C         A: 3X3 ROTATION MATRIX TO BE COMPUTED.                        ROT
C         L: 1,2 OR 3 TO ROTATE ABOUT X,Y OR Z AXIS.                    ROT
C         TH: ANGLE OF ROTATION IN RADIANS.                             ROT
C                                                                       ROT
      IMPLICIT REAL*8 (A-H,O-Z)                                         ROT
      DIMENSION  A(3,3)                                                 ROT
      COMMON/CNSNTS/ PI,RADIAN,G,THIRD,EPS(24),                         ROT
     *               UNITL,UNITM,UNITT,GRAVTY(3),TWOPI                  TWOPI
      C=DCOS(TH)                                                        ROT
      S=DSIN(TH)                                                        ROT
      IF (DABS(C) .LT.EPS(8)) C=0.0                                     CONVER
      IF (DABS(S).LT.EPS(8)) S=0.0                                      CONVER
      ONE = 1.0                                                         ROT
      IF  (1.0-DABS(C).LT.EPS(8))  C = DSIGN(ONE,C)                     CONVER
      IF  (1.0-DABS(S).LT.EPS(8))  S = DSIGN(ONE,S)                     CONVER
      IF (L.EQ.2) S = -S                                                ROT
      DO 30 I=1,3                                                       ROT
      IF(I.EQ.3)GO TO 20                                                ROT
      DO 10 J=I,2                                                       ROT
      A(I,J+1)=0.0                                                      ROT
      A(J+1,I)=0.0                                                      ROT
      IF(I+J+L.NE.5)GO TO 10                                            ROT
      A(I,J+1)=S                                                        ROT
      A(J+1,I)=-S                                                       ROT
   10 CONTINUE                                                          ROT
   20 A(I,I)= C                                                         ROT
      IF(I.EQ.L)A(I,I)=1.0                                              ROT
   30 CONTINUE                                                          ROT
      RETURN                                                            ROT
      END                                                               ROT
```

```
      SUBROUTINE ROTATE                                                 ROTATE
C                                                      REV IV    02/20/87HYPER
C     THE PURPOSE OF THIS ROUTINE IS TO TRANSFORM THOSE VARIABLES THAT  ROTATE
C     HAVE BEEN SUPPLIED IN LOCAL GEOMETRIC COORDINATES TO PRINCIPAL    ROTATE
C     AXES COORDINATES AS INDICATED BY LPMI(I) # 0 FOR I = 1 TO NSEG.   ROTATE
C                                                                       ROTATE
      IMPLICIT REAL*8 (A-H,O-Z)                                         ROTATE
      COMMON/CONTRL/ TIME,NSEG,NJNT,NPL,NBLT,NBAG,NVEH,NGRND,           ROTATE
     *               NS,NQ,NSD,NFLX,NHRNSS,NWINDF,NJNTF,NPRT(36),NPG    PAGE
      COMMON/RSAVE/  XSG(3,20,3),DPMI(3,3,30),LPMI(30),                 ATBIII
     *               NSG(9),MSG(20,9),MCG,MCGIN(24,5),KREF(20,9)        TTHKREF
      COMMON/DESCRP/ PHI(3,30),W(30),RW(30),SR(4,60),HA(3,60),HB(3,60), SLIP
     *               RPHI(3,30),HT(3,3,60),SPRING(5,90),VISC(7,90),     ROTATE
     *               JNT(30),IPIN(30),ISING(30),IGLOB(30),JOINTF(30)    ROTATE
      COMMON/CNTSRF/ PL(24,30),BELT(20,8),TPTS(6,8),BD(24,40)           EDGE
      COMMON/SGMNTS/ D(3,3,30),WMEG(3,30),WMEGD(3,30),U1(3,30),U2(3,30),ROTATE
     *               SEGLP(3,30),SEGLV(3,30),SEGLA(3,30),NSYM(30)       ROTATE
      COMMON/JBARTZ/ MNPL(    30),MNBLT(    8),MNSEG(    30),MNBAG(    6), ROTATE
     *               MPL(3,5,30),MBLT(3,5,8),MSEG(3,5,30),MBAG(3,10,6), ROTATE
     *               NTPL( 5,30),NTBLT( 5,8),NTSEG( 5,30)               ROTATE
      COMMON/CSTRNT/ A13(3,3,24),A23(3,3,24),B31(3,3,24),B32(3,3,24),   ROTATE
     *               HHT(3,3,12),RK1(3,12),RK2(3,12),QQ(3,12),TQQ(3,12),ROTATE
     *               RQQ(3,12),HQQ(3,12),SQQ(12),CFQQ(12),              ROTATE
     *               KQ1(12),KQ2(12),KQTYPE(12)                         ROTATE
      COMMON/DAMPER/ APSDM(3,20),APSDN(3,20),ASD(5,20),MSDM(20),MSDN(20)ROTATE
      COMMON/WINDFR/ WTIME(30),QFU(3,5),QFV(3,5),WF(3,30),IWIND(30),    WINDOP
     *               MWSEG(7,30),NFVSEG(6),NFVNT(5),MOWSEG(30,30)       WINDOP
      COMMON/HRNESS/ BAR(15,100),BB(100),BBDOT(100),PLOSS(2,100),       ROTATE
     *               XLONG(20),HTIME(2),IBAR(5,100),NL(2,100),          ROTATE
     *               NPTSPB(20),NPTPLY(20),NTHRNS(20),NBLTPH(5)         ROTATE
      COMMON/TABLES/MXNTI,MXNTB,MXTB1,MXTB2,NTI(50),NTAB(1250),TAB(4500)WINDROT
      COMMON/CEULER/ IEULER(30),HIR(3,3,90),ANG(3,30),ANGD(3,30),       FXHROT
     *               FE(3,30),TQE(3,30),CONST(5,30)                     FXHROT
      COMMON/TEMPVS/ T1(3),T3(3,3),LBD(40),T2(3),T4(3,3)                FXHROT
C                                                                       ROTATE
C     TRANSFORM DIRECTION COSINE MATRICEES D FROM INPUT CARDS G.3.      ROTATE
C                                                                       ROTATE
      LTEST = 0                                                         ROTATE
      DO  20  J=1,30                                                    ROTATE
      IF  (J.GT.NSEG)  LPMI(J) = 0                                      ROTATE
      IF  (LPMI(J).EQ.0)  GO TO 20                                      ROTATE
      LTEST = 1                                                         ROTATE
      DO  12  I=1,3                                                     ROTATE
      T1(I) = WMEG(I,J)                                                 ROTATE
      DO  12  K=1,3                                                     ROTATE
   12 T3(I,K) = D(I,K,J)                                                ROTATE
      CALL MAT33 (DPMI(1,1,J),T3,D(1,1,J))                              ROTATE
      CALL MAT31 (DPMI(1,1,J),T1,WMEG(1,J))                             ROTATE
   20 CONTINUE                                                          ROTATE
      IF  (LTEST.EQ.0)  GO TO 99                                        ROTATE
```

```
C                                                                       ROTATE
C     TRANSFORM SR,HT AND HB FROM INPUT CARDS B.3.                      ROTATE
C                                                                       ROTATE
      IF  (NJNT.LE.0)  GO TO 31                                         ROTATE
      DO  30  J=1,NJNT                                                  ROTATE
      I = IABS(JNT(J))                                                  ROTATE
      M = 2                                                             SLIP
      IF (IABS(IPIN(J)).GT.4) M = 3                                     SLIP
      DO  24  K=1,2                                                     ROTATE
      IF  (I.EQ.0 .OR. LPMI(I).EQ.0)  GO TO 24                          ROTATE
      IJ = 2*J-2+K                                                      ROTATE
      DO  22  LI=1,3                                                    ROTATE
      T1(LI) = SR(LI,IJ)                                                ROTATE
      T2(LI) = HB(LI,IJ)                                                FXHROT
      DO  22  LJ=1,3                                                    ROTATE
      T4(LI,LJ) = HIR(LI,LJ,IJ+30)                                      FXHROT
   22 T3(LI,LJ) = HT(LI,LJ,IJ)                                          ROTATE
      CALL MAT31 (DPMI(1,1,I),T1,SR(1,IJ))                              ROTATE
      CALL MAT31 (DPMI(1,1,I),T2,HB(1,IJ))                              FXHROT
      CALL MAT33 (DPMI(1,1,I),T3,HT(1,1,IJ))                            ROTATE
      CALL MAT33 (DPMI(1,1,I),T4,HIR(1,1,IJ+30))                        FXHROT
   24 I = J+1                                                           ROTATE
   30 CONTINUE                                                          ROTATE
C                                                                       ROTATE
C     TRANSFORM RK1,RK2 FROM INPUT CARDS D.6.                           ROTATE
C                                                                       ROTATE
   31 IF  (NQ.LE.0)  GO TO 41                                           ROTATE
      K5 = 0                                                            ROTATE
      DO  40  K=1,NQ                                                    ROTATE
      IF  (K5.EQ.1)  GO TO 39                                           ROTATE
      KSEG = KQ1(K)                                                     ROTATE
      IF  (LPMI(KSEG).EQ.0)  GO TO 36                                   ROTATE
      DO  35  I=1,3                                                     ROTATE
   35 T1(I) = RK1(I,K)                                                  ROTATE
      CALL MAT31 (DPMI(1,1,KSEG),T1,RK1(1,K))                           ROTATE
   36 KSEG = KQ2(K)                                                     ROTATE
      IF  (LPMI(KSEG).EQ.0)  GO TO 40                                   ROTATE
      DO  37  I=1,3                                                     ROTATE
   37 T1(I) = RK2(I,K)                                                  ROTATE
      CALL MAT31 (DPMI(1,1,KSEG),T1,RK2(1,K))                           ROTATE
   39 IF  (KQTYPE(K).EQ.5) K5 = 1-K5                                    ROTATE
   40 CONTINUE                                                          ROTATE
C                                                                       ROTATE
C     TRANSFORM APSDM,APSDN FROM INPUT CARDS D.8.                       ROTATE
C                                                                       ROTATE
   41 IF  (NSD.LE.0)  GO TO 151                                         FIXROT
      DO  50  J=1,NSD                                                   ROTATE
      KSEG = MSDM(J)                                                    ROTATE
      IF  (LPMI(KSEG).EQ.0)  GO TO 44                                   ROTATE
      DO  43  I=1,3                                                     ROTATE
```

```
   43 T1(I) = APSDM(I,J)                                          ROTATE
      CALL MAT31 (DPMI(1,1,KSEG),T1,APSDM(1,J))                   ROTATE
   44 KSEG = MSDN(J)                                              ROTATE
      IF  (LPMI(KSEG).EQ.0)  GO TO 50                             ROTATE
      DO  45  I=1,3                                               ROTATE
   45 T1(I) = APSDN(I,J)                                          ROTATE
      CALL MAT31 (DPMI(1,1,KSEG),T1,APSDN(1,J))                   ROTATE
   50 CONTINUE                                                    ROTATE
C                                                                 FIXROT
C     TRANSFORM QFU AND QFV FROM INPUT CARDS D.9.                 FIXROT
C                                                                 FIXROT
  151 NFORCE = NFVSEG(6)                                          FIXROT
      IF (NFORCE.LE.0) GO TO 100                                  WINDROT
      DO 152 J=1,NFORCE                                           FIXROT
      KSEG = IABS(NFVSEG(J))                                      FIXROT
      IF (LPMI(KSEG).EQ.0) GO TO 152                              FIXROT
      DO 143 I=1,3                                                FIXROT
      T1(I) = QFU(I,J)                                            FIXROT
  143 T2(I) = QFV(I,J)                                            FIXROT
      CALL MAT31 (DPMI(1,1,KSEG),T1,QFU(1,J))                     FIXROT
      CALL MAT31 (DPMI(1,1,KSEG),T2,QFV(1,J))                     FIXROT
  152 CONTINUE                                                    FIXROT
C                                                                 WINDROT
C     ROTATE WIND FORCE FUNCTIONS                                 WINDROT
C                                                                 WINDROT
  100 IF (NWINDF.EQ.0) GOTO 51                                    WINDROT
      DO 101 I=1,NSEG                                             WINDROT
      IF (MWSEG(1,I).EQ.0) GOTO 101                               WINDROT
      NT = MWSEG(5,I)                                             WINDROT
      DO 102 J=1,I-1                                              WINDROT
      IF (NT.EQ.MWSEG(5,J)) GOTO 101                              WINDROT
  102 CONTINUE                                                    WINDROT
      KT = NTI(NT)                                                WINDROT
      RK = TAB(KT)                                                WINDROT
      IF (RK.NE.0) GOTO 101                                       WINDROT
      NSR = IDINT(TAB(KT+4))                                      WINDROT
      IF (NSR.EQ.0 .OR. LPMI(NSR).EQ.0) GOTO 101                  WINDROT
      NENTRY = TAB(KT+5)                                          WINDROT
      K1 = KT+6                                                   WINDROT
      K2 = 4*NENTRY+KT+2                                          WINDROT
      DO 103 K=K1,K2,4                                            WINDROT
      DO 104 J=1,3                                                WINDROT
  104 T1(J) = TAB(K+J)                                            WINDROT
  103 CALL MAT31(DPMI(1,1,NSR),T1,TAB(K+1))                       WINDROT
  101 CONTINUE                                                    WINDROT
C                                                                 ROTATE
C     CHECK PLANE AND ELLIPSOID ASSIGNMENTS ON INPUT CARDS F.1.   ROTATE
C     TRANSFORM PLANE ARRAYS SET UP FROM INPUT CARD D.1.          ROTATE
C                                                                 ROTATE
   51 DO  52  J=1,40                                              ROTATE
```

```
      LBD(J) = 0                                                        ROTATE
   52 IF  (J.LE.NSEG)  LBD(J) = J                                       ROTATE
      IF  (NPL.LE.0)  GO TO 61                                          ROTATE
      DO  60  J=1,NPL                                                   ROTATE
      IF  (MNPL(J).EQ.0)  GO TO 60                                      ROTATE
      LPL = 0                                                           ROTATE
      KPL = MNPL(J)                                                     ROTATE
      DO  56  I=1,KPL                                                   ROTATE
      M1 = MPL (1,I,J)                                                  ROTATE
      M2 = MPL (2,I,J)                                                  ROTATE
      M3 = MPL (3,I,J)                                                  ROTATE
      IF  (LPL.EQ.M1 .OR. LPL.EQ.0)  GO TO 54                           ROTATE
      WRITE  (6,53) J,M1,LPL                                            ROTATE
   53 FORMAT('0 INPUT ERROR HAS BEEN DETECTED IN SUBROUTINE ROTATE.'/   ROTATE
     *       '  PLANE NO.',I3,' HAS BEEN ASSIGNED TO BOTH SEGMENTS NO.',ROTATE
     *       I3,' AND NO.',I3,'.'/'  PROGRAM IS BEING TERMINATED.')     ROTATE
      STOP 43                                                           ROTATE
   54 LPL = M1                                                          ROTATE
      IF  (LBD(M3).EQ.M2 .OR. LBD(M3).EQ.0)  GO TO 55                   ROTATE
      WRITE  (6,68)  M3,M2,LBD(M3)                                      ROTATE
      STOP 44                                                           ROTATE
   55 LBD(M3) = M2                                                      ROTATE
   56 CONTINUE                                                          ROTATE
      IF  (LPMI(LPL).EQ.0)  GO TO 60                                    ROTATE
      L = 1                                                             EDGE
      DO  59  K=1,6                                                     EDGE
      IF((K.EQ.3).OR.(K.EQ.6)) L = L-1                                  EDGE
      IF((K.EQ.4).OR.(K.EQ.5)) L = L+1                                  EDGE
      DO  58  I=1,3                                                     ROTATE
      T1(I) = PL(L,J)                                                   EDGE
   58 L=L+1                                                             EDGE
      CALL MAT31 (DPMI(1,1,LPL),T1,PL(L-3,J))                           EDGE
   59 L=L+1                                                             EDGE
   60 CONTINUE                                                          ROTATE
C                                                                       ROTATE
C     CHECK ELLIPSOID ASSIGNMENTS ON INPUT CARDS F.2.                   ROTATE
C     TRANSFORM BELT(L,J) FOR L=1,9 FROM INPUT CARDS D.3.               ROTATE
C                                                                       ROTATE
   61 IF  (NBLT.LE.0)  GO TO 66                                         ROTATE
      DO  65  J=1,NBLT                                                  ROTATE
      IF  (MNBLT(J).EQ.0)  GO TO 65                                     ROTATE
      KBLT = MNBLT(J)                                                   ROTATE
      DO  62  I=1,KBLT                                                  ROTATE
      M1 = MBLT(1,I,J)                                                  ROTATE
      M2 = MBLT(2,I,J)                                                  ROTATE
      M3 = MBLT(3,I,J)                                                  ROTATE
      IF  (LBD(M3).EQ.M2 .OR. LBD(M3).EQ.0)  GO TO 62                   ROTATE
      WRITE  (6,68)  M3,M2,LBD(M3)                                      ROTATE
      STOP 45                                                           ROTATE
   62 LBD(M3) = M2                                                      ROTATE
```

```
      IF  (LPMI(M1).EQ.0)  GO TO 63                                     ROTATE
      DO  57  I=1,3                                                     ROTATE
      T3(I,1) = BELT(I  ,J)                                             ROTATE
   57 T3(I,2) = BELT(I+3,J)                                             ROTATE
      CALL MAT31 (DPMI(1,1,M1),T3(1,1),BELT(1,J))                       ROTATE
      CALL MAT31 (DPMI(1,1,M1),T3(1,2),BELT(4,J))                       ROTATE
   63 IF  (LPMI(M2).EQ.0)  GO TO 65                                     ROTATE
      DO  64  I=1,3                                                     ROTATE
   64 T3(I,3) = BELT(I+6,J)                                             ROTATE
      CALL MAT31 (DPMI(1,1,M2),T3(1,3),BELT(7,J))                       ROTATE
   65 CONTINUE                                                          ROTATE
C                                                                       ROTATE
C     CHECK ELLIPSOID ASSIGNMENTS ON INPUT CARDS F.3.                   ROTATE
C                                                                       ROTATE
   66 DO  70  J=1,NSEG                                                  ROTATE
      IF  (MNSEG(J).EQ.0)  GO TO 70                                     ROTATE
      KSEG = MNSEG(J)                                                   ROTATE
      DO  69  I=1,KSEG                                                  ROTATE
      M1 = MSEG(1,I,J)                                                  ROTATE
      M2 = MSEG(2,I,J)                                                  ROTATE
      M3 = MSEG(3,I,J)                                                  ROTATE
      IF  (LBD(M1).EQ.J .OR. LBD(M1).EQ.0)  GO TO 67                    ROTATE
      WRITE  (6,68)  M1,J,LBD(M1)                                       ROTATE
      STOP 46                                                           ROTATE
   67 LBD(M1) = J                                                       ROTATE
      IF  (LBD(M3).EQ.M2 .OR. LBD(M3).EQ.0)  GO TO 69                   ROTATE
      WRITE  (6,68)  M3,M2,LBD(M3)                                      ROTATE
   68 FORMAT('0 INPUT ERROR HAS BEEN DETECTED IN SUBROUTINE ROTATE.'/   ROTATE
     *   '  ELLIPSOID NO.',I3,' HAS BEEN ASSIGNED TO BOTH SEGMENTS NO.',ROTATE
     *       I3,' AND NO.',I3,'.'/'  PROGRAM IS BEING TERMINATED.')     ROTATE
      STOP 47                                                           ROTATE
   69 LBD(M3) = M2                                                      ROTATE
   70 CONTINUE                                                          ROTATE
C                                                                       ROTATE
C     CHECK ELLIPSOID ASSIGNMENTS ON INPUT CARDS F.6.                   ROTATE
C                                                                       ROTATE
      IF  (NBAG.EQ.0)  GO TO 174                                        TGMOD8
      DO  73  J=1,NBAG                                                  ROTATE
      IF  (MNBAG(J).EQ.0)  GO TO 73                                     ROTATE
      KBAG = MNBAG(J)                                                   ROTATE
      DO  72  I=1,KBAG                                                  ROTATE
      M2 = MBAG(2,I,J)                                                  ROTATE
      M3 = MBAG(3,I,J)                                                  ROTATE
      IF  (LBD(M3).EQ.M2 .OR. LBD(M3).EQ.0)  GO TO 72                   ROTATE
      WRITE  (6,68)  M3,M2,LBD(M3)                                      ROTATE
      STOP 50                                                           ROTATE
   72 LBD(M3) = M2                                                      ROTATE
   73 CONTINUE                                                          ROTATE
C                                                                       TGMOD8
C     CHECK ELLIPSOID ASSIGNMENTS ON INPUT CARDS F.7.                   TGMOD8
```

```
C                                                                       TGMOD8
  174 IF(NWINDF.EQ.0) GO TO 74                                          TGMOD8
      DO 175 J=1,NSEG                                                   TGMOD8
       M1 = IABS(MWSEG(1,J))                                            TGMOD8
       IF(M1.EQ.0) GO TO 175                                            TGMOD8
       M2 = MWSEG(2,J)                                                  TGMOD8
       IF(LBD(M2).EQ.M1.OR.LBD(M2).EQ.0) GO TO 172                      TGMOD8
       WRITE(6,68) M2,M1,LBD(M2)                                        TGMOD8
       STOP 48                                                          TGMOD8
  172  LBD(M2) = M1                                                     TGMOD8
  175 CONTINUE                                                          TGMOD8
C                                                                       ROTATE
C     CHECK ELLIPSOID ASSIGNMENTS ON INPUT CARDS F.8.                   ROTATE
C     TRANSFORM BAR(L,K) FOR L=4,12 FROM INPUT CARDS F.8.D.             ROTATE
C                                                                       ROTATE
   74 IF  (NHRNSS.EQ.0)  GO TO 81                                       ROTATE
      J1 = 1                                                            ROTATE
      K1 = 1                                                            ROTATE
      DO  80  II=1,NHRNSS                                               ROTATE
      IF  (NBLTPH(II).LE.0)  GO TO 80                                   ROTATE
      J2 = J1 + NBLTPH(II) - 1                                          ROTATE
      DO  79  JJ=J1,J2                                                  ROTATE
      IF  (NPTSPB(JJ).LE.0)  GO TO 79                                   ROTATE
      K2 = K1 + NPTSPB(JJ) - 1                                          ROTATE
      DO  78  K=K1,K2                                                   ROTATE
      M2 = MOD(IBAR(1,K),100)                                           ROTATE
      M3 = IBAR(2,K)                                                    ROTATE
      IF (M3.EQ.0) GO TO 88                                             BUTLER1
      IF  (LBD(M3).EQ.M2 .OR. LBD(M3).EQ.0)  GO TO 75                   ROTATE
      WRITE  (6,68)  M3,M2,LBD(M3)                                      ROTATE
      STOP 51                                                           ROTATE
   75 LBD(M3) = M2                                                      ROTATE
   88 IF  (LPMI(M2).EQ.0)  GO TO 78                                     BUTLER1
      DO  77  J=3,9,3                                                   ROTATE
      DO  76  I=1,3                                                     ROTATE
      IJ = I+J                                                          ROTATE
   76 T1(I) = BAR(IJ,K)                                                 ROTATE
   77 CALL MAT31 (DPMI(1,1,M2),T1,BAR(J+1,K))                           ROTATE
   78 CONTINUE                                                          ROTATE
      K1 = K2+1                                                         ROTATE
   79 CONTINUE                                                          ROTATE
      J1 = J2+1                                                         ROTATE
   80 CONTINUE                                                          ROTATE
C                                                                       ROTATE
C     TRANSFORM DATA IN BD ARRAYS FOR ELLIPSOIDS THAT HAVE BEEN ASSIGNEDROTATE
C                                                                       ROTATE
   81 DO  90  J=1,40                                                    ROTATE
      IF  (LBD(J).EQ.0)  GO TO 90                                       ROTATE
      KSEG = LBD(J)                                                     ROTATE
      IF  (LPMI(KSEG).EQ.0)  GO TO 90                                   ROTATE
```

```
      L = 4                                                   HYPER
      IF (BD(1,J).LT.0.0) L = 5                               HYPER
      M = 8                                                   HYPER
      DO  82  I=1,3                                           ROTATE
      T1(I) = BD(L,J)                                         HYPER
      L = L + 1                                               HYPER
      DO 82 K = 1,3                                           HYPER
      T3(K,I) = BD(M,J)                                       HYPER
   82 M = M + 1                                               HYPER
      CALL MAT31 (DPMI(1,1,KSEG),T1,BD(L-3,J))                HYPER
      IF (BD(1,J).GT.0.0) GO TO 84                            HYPER
      CALL MAT33 (DPMI(1,1,KSEG),T3,BD(8,J))                  HYPER
      GO TO 90                                                HYPER
   84 CALL DOTT33 (BD( 7,J),DPMI(1,1,KSEG),T3)                HYPER
      CALL MAT33  (DPMI(1,1,KSEG),T3,BD( 7,J))                ROTATE
      CALL DOTT33 (BD(16,J),DPMI(1,1,KSEG),T3)                ROTATE
      CALL MAT33  (DPMI(1,1,KSEG),T3,BD(16,J))                ROTATE
   90 CONTINUE                                                ROTATE
   99 RETURN                                                  ROTATE
      END                                                     ROTATE
```

```
      SUBROUTINE RSTART(IF,IT)                                          RSTART
C                                                  REV IV    07/24/86SLIP
C     THE FIVE FUNCTIONS OF SUBROUTINE RSTART ARE:                      RSTART
C        1. READ INPUT & INITIALIZATION RECORD FROM OLD RESTART TAPE.   RSTART
C        2. WRITE INPUT & INITIALIZATION RECORD ONTO NEW RESTART TAPE.  RSTART
C        3. READ TIME POINT RECORD FROM OLD RESTART TAPE.               RSTART
C        4. READ NEW INPUT DATA FROM INPUT STREAM FOR RESTART.          RSTART
C        5. WRITE TIME POINT RECORD ONTO NEW RESTART TAPE.              RSTART
C                                                                       RSTART
      IMPLICIT REAL*8(A-H,O-Z)                                          RSTART
C                                                                       RSTART
C     ALL LABELED COMMON BLOCKS ARE INCLUDED HERE                       RSTART
C     TO GIVE A COMPLETE SET FOR REFERENCE                              RSTART
C  1                                                                    RSTART
      COMMON/CONTRL/ TIME,NSEG,NJNT,NPL,NBLT,NBAG,NVEH,NGRND,           RSTART
     *               NS,NQ,NSD,NFLX,NHRNSS,NWINDF,NJNTF,NPRT(36),NPG    PAGE
      DIMENSION IC1(51)                                                 PAGE
      EQUIVALENCE (IC1(1),NSEG)                                         RSTART
C  2                                                                    RSTART
      COMMON/CNTSRF/ PL(24,30),BELT(20,8),TPTS(6,8),BD(24,40)           EDGE
      DIMENSION RC2(1888)                                               EDGE
      EQUIVALENCE (RC2(1),PL(1,1))                                      RSTART
C  3                                                                    RSTART
      COMMON/VPOSTN/ ZPLT(3),SPLT(3),AXV(3,6),VATAB(6,501,6),           VEHICL
     *               VT0(6),VDT(6),TIMEV(6),OMEGV(6),NVTAB(6),INDXV(6)  RSTART
      DIMENSION RC3(18084),IC3(12)                                      VEHICL
      EQUIVALENCE (RC3(1),ZPLT(1)),(IC3(1),NVTAB(1))                    RSTART
C  4                                                                    RSTART
      COMMON/SGMNTS/ D(3,3,30),WMEG(3,30),WMEGD(3,30),U1(3,30),U2(3,30),RSTART
     *               SEGLP(3,30),SEGLV(3,30),SEGLA(3,30),NSYM(30)       RSTART
      DIMENSION RC4(900)                                                RSTART
      EQUIVALENCE (RC4(1),D(1,1,1))                                     RSTART
C  5                                                                    RSTART
      COMMON/CMATRX/ V1(3,30),V2(3,30),V3(3,12),B12(3,3,60),A22(3,3,60),RSTART
     *               F(3,30),TQ(3,30),WJ(30),A11(3,3,30)                SLIP
      DIMENSION RC5A(1296),RC5B(480)                                    SLIP
      EQUIVALENCE (RC5A(1),V1(1,1)),(RC5B(1),F(1,1))                    RSTART
C  6                                                                    RSTART
      COMMON/ABDATA/ ZDEP(3,5),DBR(3,3,5),DPVCTR(3,5),DEPLOY(3,5),      RSTART
     *               AB(3,5),B(9,4,5),ZR(3,4,5),BFB(3,4,5),DRR(9,4,5),  RSTART
     *               VBAGG(5),VSCS(5),SPRK(5),CK(5),CMASS(5),CYMIN(5),  RSTART
     *               CYMOUT(5),BAGPV(5),PD(5),VBAG(5),VOLBP(5),         RSTART
     *               PCYV(5),PCYMIN(5),PVBAG(5),TV1(3,4,5),TV2(3,10,5), RSTART
     *               SWITCH(5),PYMOUT(5),SCALE(5),PREVT,IFULL(6)        RSTART
      DIMENSION RC6A(610),RC6B(271)                                     RSTART
      EQUIVALENCE (RC6A(1),ZDEP(1,1)) , (RC6B(1),CYMIN(1))              RSTART
C  7                                                                    RSTART
      COMMON/TITLES/ DATE(3),COMENT(40),VPSTTL(20),BDYTTL(5),           RSTART
     *               BLTTTL(5,8),PLTTL(5,30),BAGTTL(5,6),SEG(30),       RSTART
     *               JOINT(30),CGS(30),JS(30)                           RSTART
```

```
      REAL DATE,COMENT,VPSTTL,BDYTTL,BLTTTL,PLTTL,BAGTTL,SEG,JOINT        RSTART
      LOGICAL*1 CGS,JS                                                    RSTART
      REAL  RC7,RC7A,XDTE,XCMENT                                          RSTART
      DIMENSION RC7(305),RC7A(348),XDTE(3),XCMENT(40)                     RSTART
      EQUIVALENCE (RC7(1),VPSTTL(1)),(RC7A(1),DATE(1))                    RSTART
C  8                                                                      RSTART
      COMMON/CNSNTS/ PI,RADIAN,G,THIRD,EPS(24),                           RSTART
     *               UNITL,UNITM,UNITT,GRAVTY(3),TWOPI                    TWOPI
      DIMENSION RC8(35)                                                   TWOPI
      EQUIVALENCE (RC8(1),PI)                                             RSTART
C  9                                                                      RSTART
      COMMON/DESCRP/ PHI(3,30),W(30),RW(30),SR(4,60),HA(3,60),HB(3,60),   SLIP
     *               RPHI(3,30),HT(3,3,60),SPRING(5,90),VISC(7,90),       RSTART
     *               JNT(30),IPIN(30),ISING(30),IGLOB(30),JOINTF(30)      RSTART
      DIMENSION RC9(2460),IC9(150)                                        SLIP
      EQUIVALENCE (RC9(1),PHI(1,1)),(IC9(1),JNT(1))                       RSTART
C 10                                                                      RSTART
      COMMON/JBARTZ/ MNPL(   30),MNBLT(   8),MNSEG(   30),MNBAG(    6),   RSTART
     *               MPL(3,5,30),MBLT(3,5,8),MSEG(3,5,30),MBAG(3,10,6),   RSTART
     *               NTPL( 5,30),NTBLT( 5,8),NTSEG( 5,30)                 RSTART
      DIMENSION IC10(1614)                                                RSTART
      EQUIVALENCE (IC10(1),MNPL(1))                                       RSTART
C 11                                                                      RSTART
      COMMON/FORCES/PSF(7,70),BSF(4,20),SSF(10,40),BAGSF(3,20),           NCFORC
     *               PRJNT(7,30),NPANEL(5),NPSF,NBSF,NSSF,NBGSF           RSTART
      DIMENSION RC11(1240),IC11(9)                                        NCFORC
      EQUIVALENCE (RC11(1),PSF(1,1)),(IC11(1),NPANEL(1))                  RSTART
C 12                                                                      RSTART
      COMMON/INTEST/ SGTEST(3,4,30),XTEST(3,120),SEGT(120),REGT(120)      RSTART
      REAL    SEGT                                                        RSTART
      DIMENSION RC12(720)                                                 RSTART
      EQUIVALENCE (RC12(1),SGTEST(1,1,1))                                 RSTART
C 13                                                                      RSTART
      COMMON/CSTRNT/ A13(3,3,24),A23(3,3,24),B31(3,3,24),B32(3,3,24),     RSTART
     *               HHT(3,3,12),RK1(3,12),RK2(3,12),QQ(3,12),TQQ(3,12),RSTART
     *               RQQ(3,12),HQQ(3,12),SQQ(12),CFQQ(12),                RSTART
     *               KQ1(12),KQ2(12),KQTYPE(12)                           RSTART
      DIMENSION RC13(72),IC13(36),RC13A(1212),RC13H(348)                  RSTART
      EQUIVALENCE (RC13(1),RK1(1,1)),(IC13(1),KQ1(1)),                    RSTART
     *            (RC13A(1),A13(1,1,1)),(RC13H(1),HHT(1,1,1))             RSTART
C 14                                                                      RSTART
      COMMON/TABLES/MXNTI,MXNTB,MXTB1,MXTB2,NTI(50),NTAB(1250),TAB(4500)DIMENB
      DIMENSION IC14(1304)                                                BUTLER2
      EQUIVALENCE (IC14(1),MXNTI)                                         RSTART
C 15                                                                      RSTART
      COMMON/COMAIN/VAR(240),DER(240),DT,H0,HMAX,HMIN,RSTIME,             RSTART
     *             ISTEP,NSTEPS,NDINT,NEQ,IRSIN,IRSOUT                    RSTART
      DIMENSION RC15(485),IC15(6)                                         RSTART
      EQUIVALENCE (RC15(1),VAR(1)),(IC15(1),ISTEP)                        RSTART
C 16                                                                      RSTART
```

```
      COMMON/CDINT/ UU(4),GH(3,4),                                       RSTART
     *              E(3,240),FF(5,240),GG(5,240),Y(5,240),U(5,240),      RSTART
     *              H,HPRINT,HS,TPRINT,TSTART,ICNT,IDBL,IFLAG            RSTART
C      NOTE: FF REPLACES F FROM SUBROUTINE DINT.                         RSTART
      DIMENSION RC16(5541),IC16(3)                                       RSTART
      EQUIVALENCE (RC16(1),UU(1)),(IC16(1),ICNT)                         RSTART
C 17                                                                     RSTART
      COMMON/DAMPER/ APSDM(3,20),APSDN(3,20),ASD(5,20),MSDM(20),MSDN(20)RSTART
      DIMENSION RC17(220),IC17(40)                                       RSTART
      EQUIVALENCE (RC17(1),APSDM(1,1)),(IC17(1),MSDM(1))                 RSTART
C 18                                                                     RSTART
      COMMON/CEULER/ IEULER(30),HIR(3,3,90),ANG(3,30),ANGD(3,30),        JDRIFT
     *               FE(3,30),TQE(3,30),CONST(5,30)                      JDRIFT
      DIMENSION RC18(1320)                                               JDRIFT
      EQUIVALENCE (RC18(1),HIR(1,1,1))                                   RSTART
C 19                                                                     RSTART
      COMMON/TEMPVI/ CREST,TTI(3),R1I(3),R2I(3),JSTOP(4,2,30)            RSTART
      DIMENSION RC19(10),IC19(180)                                       RSTART
      EQUIVALENCE (RC19(1),CREST),(IC19(1),JSTOP(1,1,1))                 RSTART
C 20                                                                     RSTART
      COMMON/CYDATA/ CYTD(5),CYPA(5),CYSP(5),CYTO(5),CYVO(5),CYCD(5),    RSTART
     *               CYK(5),CYR(5),CYAT(5),CYPV(5),CYCDO(5),CYAO(5),     RSTART
     *               CYPO(5),CYSS(5),CYLO(5),CYC(5),CYRHOO(5),CYVMAX(5),RSTART
     *               CYORFC(5),CYRHO(5),CYT(5),CYP(5),CYV(5)             RSTART
      DIMENSION RC20A(95),RC20B(20)                                      RSTART
      EQUIVALENCE (RC20A(1),CYTD(1)) , (RC20B(1),CYRHO(1))               RSTART
C 21                                                                     RSTART
      COMMON/RSAVE/  XSG(3,20,3),DPMI(3,3,30),LPMI(30),                  ATBIII
     *               NSG(9),MSG(20,9),MCG,MCGIN(24,5),KREF(20,9)         TTHKREF
      DIMENSION RC21(450),IC21(520)                                      TTHKREF
      EQUIVALENCE (RC21(1),XSG(1,1,1)) , (IC21(1),LPMI(1))               RSTART
C 22                                                                     RSTART
      COMMON/FLXBLE/ HF(4,12,8),B42(3,3,24),V4(3,8),NFLEX(3,8)           RSTART
      DIMENSION RC22(624),IC22(24)                                       RSTART
      EQUIVALENCE (RC22(1),HF(1,1,1)) , (IC22(1),NFLEX(1,1))             RSTART
C 23                                                                     RSTART
      COMMON/HRNESS/ BAR(15,100),BB(100),BBDOT(100),PLOSS(2,100),        RSTART
     *               XLONG(20),HTIME(2),IBAR(5,100),NL(2,100),           RSTART
     *               NPTSPB(20),NPTPLY(20),NTHRNS(20),NBLTPH(5)          RSTART
      DIMENSION RC23(1922),IC23(765)                                     RSTART
      EQUIVALENCE (RC23(1),BAR(1,1)),(IC23(1),IBAR(1,1))                 RSTART
C 24                                                                     RSTART
      COMMON/WINDFR/ WTIME(30),QFU(3,5),QFV(3,5),WF(3,30),IWIND(30),     WINDOP
     *               MWSEG(7,30),NFVSEG(6),NFVNT(5),MOWSEG(30,30)        WINDOP
      DIMENSION RC24(150),IC24(1151)                                     WINDOP
      EQUIVALENCE (RC24(1),WTIME(1)),(IC24(1),IWIND(1))                  RSTART
C                                                                        RSTART
      REAL AOLD4,AAOLD4                                                  RSTART
      DIMENSION COMMON(24),INDEX(3)                                      RSTART
      DATA COMMON /8HCONTRL  ,8HCNTSRF  ,8HVPOSTN  ,8HSGMNTS  ,          RSTART
```

```
     *              8HCMATRX  ,8HABDATA  ,8HTITLES  ,8HCNSNTS  ,        RSTART
     *              8HDESCRP  ,8HJBARTZ  ,8HFORCES  ,8HINTEST  ,        RSTART
     *              8HCSTRNT  ,8HTABLES  ,8HCOMAIN  ,8HCDINT   ,        RSTART
     *              8HDAMPER  ,8HCEULER  ,8HTEMPVI  ,8HCYDATA  ,        RSTART
     *              8HRSAVE   ,8HFLXBLE  ,8HHRNESS  ,8HWINDFR  /        RSTART
      DATA BLANK/8H        /                                            RSTART
      CALL ELTIME(1,25)                                                 RSTART
      GO TO (100,200,300,400,500),IF                                    RSTART
C                                                                       RSTART
C        1. READ INPUT & INITIALIZATION RECORD FROM OLD RESTART TAPE.   RSTART
C                                                                       RSTART
  100 READ  (IT) IC1, PL, RC3, IC3, NSYM, RC6A, IFULL, XDTE, XCMENT,    RSTART
     *           RC7, CGS, JS, RC8, RC9, IC9, IC10, NPANEL, SGTEST,     RSTART
     *           RC13, IC13, IC14, DT, H0, HMAX, HMIN, NSTEPS, NDINT,   RSTART
     *           RC17, IC17, IEULER, IC19, RC20A, RC21, IC21, HF, IC22, RSTART
     *           RC23, IC23, RC24, IC24                                 RSTART
      WRITE (6,101) IT,XDTE,XCMENT                                      RSTART
  101 FORMAT('0 INPUT DATA HAS BEEN READ IN FROM UNIT NO.',I4//         RSTART
     *       10X,3A4//10X,20A4/10X,20A4)                                RSTART
      GO TO 999                                                         RSTART
C                                                                       RSTART
C        2. WRITE INPUT & INITIALIZATION RECORD ONTO NEW RESTART TAPE.  RSTART
C                                                                       RSTART
  200 WRITE (IT) IC1, PL, RC3, IC3, NSYM, RC6A, IFULL, DATE, COMENT,    RSTART
     *           RC7, CGS, JS, RC8, RC9, IC9, IC10, NPANEL, SGTEST,     RSTART
     *           RC13, IC13, IC14, DT, H0, HMAX, HMIN, NSTEPS, NDINT,   RSTART
     *           RC17, IC17, IEULER, IC19, RC20A, RC21, IC21, HF, IC22, RSTART
     *           RC23, IC23, RC24, IC24                                 RSTART
      GO TO 999                                                         RSTART
C                                                                       RSTART
C        3. READ TIME POINT RECORD FROM OLD RESTART TAPE.               RSTART
C                                                                       RSTART
  300 READ  (IT) TIME, BELT, TPTS, BD, RC4, RC5B, RC6B, IFULL, IPIN,    RSTART
     *           RC11, IC11, XTEST, SEGT, REGT, RC13H, KQTYPE, TAB,     RSTART
     *           VAR, DER, NEQ, RC16, IC16, IEULER, RC18, IC19, RC20B,  RSTART
     *           RC21, IC21, V4, RC23, NL, NPTPLY, WTIME, IWIND         RSTART
      CALL OUTPUT(1)                                                    RSTART
      GO TO 999                                                         RSTART
C                                                                       RSTART
C        5. WRITE TIME POINT RECORD ONTO NEW RESTART TAPE.              RSTART
C                                                                       RSTART
  500 WRITE (IT) TIME, BELT, TPTS, BD, RC4, RC5B, RC6B, IFULL, IPIN,    RSTART
     *           RC11, IC11, XTEST, SEGT, REGT, RC13H, KQTYPE, TAB,     RSTART
     *           VAR, DER, NEQ, RC16, IC16, IEULER, RC18, IC19, RC20B,  RSTART
     *           RC21, IC21, V4, RC23, NL, NPTPLY, WTIME, IWIND         RSTART
      GO TO 999                                                         RSTART
C                                                                       RSTART
C        4. READ NEW INPUT DATA FROM INPUT STREAM FOR RESTART.          RSTART
C                                                                       RSTART
  400 READ (5,399) AVAR,INDEX,ITYPE,RR,II,AA,RROLD,IIOLD,AAOLD          RSTART
```

```
  399 FORMAT(A8,4I4,2(F8.0,I8,A8))                                      RSTART
      CALL SEARCH(AVAR,INDEX,NCOM,ITEM)                                 RSTART
      IF (NCOM.LE.0)    GO TO 490                                       RSTART
      IF (NCOM.GT.24)   GO TO 999                                       RSTART
      IF (ITYPE.GT.3)   GO TO 490                                       RSTART
      GO TO ( 1,  2,  3,  4,  5,  6,  7,  8,  9, 10, 11, 12,            RSTART
     *        13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24) , NCOM    RSTART
C     COMMON /CONTRL/                                                   RSTART
    1 IF (ITEM.GT.1)    GO TO 401                                       RSTART
      IF (ITYPE.NE.1)   GO TO 490                                       RSTART
      ROLD = TIME                                                       RSTART
      TIME = RR                                                         RSTART
      GO TO 492                                                         RSTART
  401 IF (ITEM.GT.52)   GO TO 490                                       PAGE
      IF (ITYPE.NE.2)   GO TO 490                                       RSTART
      IOLD = IC1(ITEM-1)                                                RSTART
      IC1(ITEM-1) = II                                                  RSTART
      GO TO 494                                                         RSTART
C     COMMON /CNTSRF/                                                   RSTART
    2 IF (ITEM.GT.1888) GO TO 490                                       EDGE
      IF (ITYPE.NE.1)   GO TO 490                                       RSTART
      ROLD = RC2(ITEM)                                                  RSTART
      RC2(ITEM) = RR                                                    RSTART
      GO TO 492                                                         RSTART
C     COMMON /VPOSTN/                                                   RSTART
    3 IF (ITEM.GT.18084) GO TO 403                                      VEHICL
      IF (ITYPE.NE.1)   GO TO 490                                       RSTART
      ROLD = RC3(ITEM)                                                  RSTART
      RC3(ITEM) = RR                                                    RSTART
      GO TO 492                                                         RSTART
  403 IF (ITEM.GT.18096) GO TO 490                                      VEHICL
      IF (ITYPE.NE.2)   GO TO 490                                       RSTART
      IOLD = IC3(ITEM-18084)                                            VEHICL
      IC3(ITEM-18084) = II                                              VEHICL
      GO TO 494                                                         RSTART
C     COMMON /SGMNTS/                                                   RSTART
    4 IF (ITEM.GT.900 ) GO TO 404                                       RSTART
      IF (ITYPE.NE.1)   GO TO 490                                       RSTART
      ROLD = RC4(ITEM)                                                  RSTART
      RC4(ITEM) = RR                                                    RSTART
      GO TO 492                                                         RSTART
  404 IF (ITEM.GT.930 ) GO TO 490                                       RSTART
      IF (ITYPE.NE.2)   GO TO 490                                       RSTART
      IOLD = NSYM(ITEM-900)                                             RSTART
      NSYM(ITEM-900) = II                                               RSTART
      GO TO 494                                                         RSTART
C     COMMON /CMATRX/                                                   RSTART
    5 IF (ITEM.GT.1776) GO TO 490                                       SLIP
      IF (ITYPE.NE.1)   GO TO 490                                       RSTART
      ROLD = RC5A(ITEM)                                                 RSTART
```

```
      RC5A(ITEM) = RR                                                   RSTART
      GO TO 492                                                         RSTART
C     COMMON /ABDATA/                                                   RSTART
    6 IF (ITEM.GT.881 ) GO TO 406                                       RSTART
      IF (ITYPE.NE.1)   GO TO 490                                       RSTART
      ROLD = RC6A(ITEM)                                                 RSTART
      RC6A(ITEM) = RR                                                   RSTART
      GO TO 492                                                         RSTART
  406 IF (ITEM.GT.887 ) GO TO 490                                       RSTART
      IF (ITYPE.NE.2)   GO TO 490                                       RSTART
      IOLD = IFULL(ITEM-881)                                            RSTART
      IFULL(ITEM-881) = II                                              RSTART
      GO TO 494                                                         RSTART
C     COMMON /TITLES/ NOTE: NO PROVISION FOR CGS OR JS.                 RSTART
    7 IF (ITEM.GT.348 ) GO TO 490                                       RSTART
      IF (ITYPE.NE.3)   GO TO 490                                       RSTART
      AOLD = RC7A(ITEM)                                                 RSTART
      RC7A(ITEM) = AA                                                   RSTART
      GO TO 496                                                         RSTART
C     COMMON /CNSNTS/                                                   RSTART
    8 IF (ITEM.GT.35  ) GO TO 490                                       TWOPI
      IF (ITEM.GT.31  ) GO TO 408                                       RSTART
      IF (ITEM.LE.28  ) GO TO 408                                       RSTART
      IF (ITYPE.NE.3)   GO TO 490                                       RSTART
      AOLD = RC8(ITEM)                                                  RSTART
      RC8(ITEM) = AA                                                    RSTART
      GO TO 496                                                         RSTART
  408 IF (ITYPE.NE.1)   GO TO 490                                       RSTART
      ROLD = RC8(ITEM)                                                  RSTART
      RC8(ITEM) = RR                                                    RSTART
      GO TO 492                                                         RSTART
C     COMMON /DESCRP/                                                   RSTART
    9 IF (ITEM.GT.2460) GO TO 409                                       SLIP
      IF (ITYPE.NE.1)   GO TO 490                                       RSTART
      ROLD = RC9(ITEM)                                                  RSTART
      RC9(ITEM) = RR                                                    RSTART
      GO TO 492                                                         RSTART
  409 IF (ITEM.GT.2610) GO TO 490                                       SLIP
      IF (ITYPE.NE.2)   GO TO 490                                       RSTART
      IOLD = IC9(ITEM-2460)                                             SLIP
      IC9(ITEM-2460) = II                                               SLIPRT
      GO TO 494                                                         RSTART
C     COMMON /JBARTZ/                                                   RSTART
   10 IF (ITEM.GT.1614) GO TO 490                                       RSTART
      IF (ITYPE.NE.2)   GO TO 490                                       RSTART
      IOLD = IC10(ITEM)                                                 RSTART
      IC10(ITEM) = II                                                   RSTART
      GO TO 494                                                         RSTART
C     COMMON /FORCES/                                                   RSTART
   11 IF (ITEM.GT.1240) GO TO 411                                       NCFORC
```

```
      IF (ITYPE.NE.1)   GO TO 490                                       RSTART
      ROLD = RC11(ITEM)                                                 RSTART
      RC11(ITEM) = RR                                                   RSTART
      GO TO 492                                                         RSTART
  411 IF (ITEM.GT.1249) GO TO 490                                       NCFORC
      IF (ITYPE.NE.2)   GO TO 490                                       RSTART
      IOLD = IC11(ITEM-1240)                                            NCFORC
      IC11(ITEM-1240) = II                                              NCFORC
      GO TO 494                                                         RSTART
C     COMMON /INTEST/                                                   RSTART
   12 IF (ITEM.GT.720 ) GO TO 412                                       RSTART
      IF (ITYPE.NE.1)   GO TO 490                                       RSTART
      ROLD = RC12(ITEM)                                                 RSTART
      RC12(ITEM) = RR                                                   RSTART
      GO TO 492                                                         RSTART
  412 IF (ITEM.GT.840 ) GO TO 512                                       RSTART
      IF (ITYPE.NE.3)   GO TO 490                                       RSTART
      AOLD = SEGT(ITEM-720)                                             RSTART
      SEGT(ITEM-720) = AA                                               RSTART
      GO TO 496                                                         RSTART
  512 IF (ITEM.GT.960 ) GO TO 490                                       RSTART
      IF (ITYPE.NE.3)   GO TO 490                                       RSTART
      AOLD = REGT(ITEM-840)                                             RSTART
      REGT(ITEM-840) = AA                                               RSTART
      GO TO 496                                                         RSTART
C     COMMON /CSTRNT/                                                   RSTART
   13 IF (ITEM.GT.1212) GO TO 413                                       RSTART
      IF (ITYPE.NE.1)   GO TO 490                                       RSTART
      ROLD = RC13A(ITEM)                                                RSTART
      RC13A(ITEM) = RR                                                  RSTART
      GO TO 492                                                         RSTART
  413 IF (ITEM.GT.1248) GO TO 490                                       RSTART
      IF (ITYPE.NE.2)   GO TO 490                                       RSTART
      IOLD = IC13(ITEM-1212)                                            RSTART
      IC13(ITEM-1212) = II                                              RSTART
      GO TO 494                                                         RSTART
C     COMMON /TABLES/                                                   RSTART
   14 IF (ITEM.GT.1304 ) GO TO 414                                      BUTLER2
      IF (ITYPE.NE.2)   GO TO 490                                       RSTART
      IOLD = IC14(ITEM)                                                 RSTART
      IC14(ITEM) = II                                                   RSTART
      GO TO 494                                                         RSTART
  414 IF (ITEM.GT.5804) GO TO 490                                       MISC
      IF (ITYPE.NE.1)   GO TO 490                                       RSTART
      ROLD = TAB(ITEM-1304)                                             BUTLER2
      TAB(ITEM-1304) = RR                                               BUTLER2
      GO TO 492                                                         RSTART
C     COMMON /COMAIN/                                                   RSTART
   15 IF (ITEM.GT.485 ) GO TO 415                                       RSTART
      IF (ITYPE.NE.1)   GO TO 490                                       RSTART
```

```
      ROLD = RC15(ITEM)                                                 RSTART
      RC15(ITEM) = RR                                                   RSTART
      GO TO 492                                                         RSTART
  415 IF (ITEM.GT.491 ) GO TO 490                                       RSTART
      IF (ITYPE.NE.2)   GO TO 490                                       RSTART
      IOLD = IC15(ITEM-485)                                             RSTART
      IC15(ITEM-485) = II                                               RSTART
      GO TO 494                                                         RSTART
C     COMMON /CDINT /                                                   RSTART
   16 IF (ITEM.GT.5541) GO TO 416                                       RSTART
      IF (ITYPE.NE.1)   GO TO 490                                       RSTART
      ROLD = RC16(ITEM)                                                 RSTART
      RC16(ITEM) = RR                                                   RSTART
      GO TO 492                                                         RSTART
  416 IF (ITEM.GT.5544) GO TO 490                                       RSTART
      IF (ITYPE.NE.2)   GO TO 490                                       RSTART
      IOLD = IC16(ITEM-5541)                                            RSTART
      IC16(ITEM-5541) = II                                              RSTART
      GO TO 494                                                         RSTART
C     COMMON /DAMPER/                                                   RSTART
   17 IF (ITEM.GT.220 ) GO TO 417                                       RSTART
      IF (ITYPE.NE.1)   GO TO 490                                       RSTART
      ROLD = RC17(ITEM)                                                 RSTART
      RC17(ITEM) = RR                                                   RSTART
      GO TO 492                                                         RSTART
  417 IF (ITEM.GT.260 ) GO TO 490                                       RSTART
      IF (ITYPE.NE.2)   GO TO 490                                       RSTART
      IOLD = IC17(ITEM-220)                                             RSTART
      IC17(ITEM-220) = II                                               RSTART
      GO TO 494                                                         RSTART
C     COMMON /CEULER/                                                   RSTART
   18 IF (ITEM.GT.30   ) GO TO 418                                      RSTART
      IF (ITYPE.NE.2)   GO TO 490                                       RSTART
      IOLD = IEULER(ITEM)                                               RSTART
      IEULER(ITEM) = II                                                 RSTART
      GO TO 494                                                         RSTART
  418 IF (ITEM.GT.1350) GO TO 490                                       JDRIFT
      IF (ITYPE.NE.1)   GO TO 490                                       RSTART
      ROLD = RC18(ITEM-30)                                              RSTART
      RC18(ITEM-30) = RR                                                RSTART
      GO TO 492                                                         RSTART
C     COMMON /TEMPVI/                                                   RSTART
   19 IF (ITEM.GT.10   ) GO TO 419                                      RSTART
      IF (ITYPE.NE.1)   GO TO 490                                       RSTART
      ROLD = RC19(ITEM)                                                 RSTART
      RC19(ITEM) = RR                                                   RSTART
      GO TO 492                                                         RSTART
  419 IF (ITEM.GT.190 ) GO TO 490                                       RSTART
      IF (ITYPE.NE.2)   GO TO 490                                       RSTART
      IOLD = IC19(ITEM-10)                                              RSTART
```

```
      IC19(ITEM-10) = II                                              RSTART
      GO TO 494                                                       RSTART
C     COMMON/CYDATA/                                                  RSTART
   20 IF (ITEM.GT.115 ) GO TO 490                                     RSTART
      IF (ITYPE.NE.1)   GO TO 490                                     RSTART
      ROLD = RC20A(ITEM)                                              RSTART
      RC20A(ITEM) = RR                                                RSTART
      GO TO 492                                                       RSTART
C     COMMON /RSAVE/                                                  RSTART
   21 IF (ITEM.GT.450 ) GO TO 421                                     RSTART
      IF (ITYPE.NE.1  ) GO TO 490                                     RSTART
      ROLD = RC21(ITEM)                                               RSTART
      RC21(ITEM) = RR                                                 RSTART
      GO TO 492                                                       RSTART
  421 IF (ITEM.GT.970 ) GO TO 490                                     TTHKREF
      IF (ITYPE.NE.2  ) GO TO 490                                     RSTART
      IOLD = IC21(ITEM-450)                                           RSTART
      IC21(ITEM-450) = II                                             RSTART
      GO TO 494                                                       RSTART
C     COMMON /FLXBLE/                                                 RSTART
   22 IF (ITEM.GT.624 ) GO TO 422                                     RSTART
      IF (ITYPE.NE.1  ) GO TO 490                                     RSTART
      ROLD = RC22(ITEM)                                               RSTART
      RC22(ITEM) = RR                                                 RSTART
      GO TO 492                                                       RSTART
  422 IF (ITEM.GT.648 ) GO TO 490                                     RSTART
      IF (ITYPE.NE.2  ) GO TO 490                                     RSTART
      IOLD = IC22(ITEM-624)                                           RSTART
      IC22(ITEM-624) = II                                             RSTART
      GO TO 494                                                       RSTART
C     COMMON /HRNESS/                                                 RSTART
   23 IF (ITEM.GT.1922) GO TO 423                                     RSTART
      IF (ITYPE.NE.1)   GO TO 490                                     RSTART
      ROLD = RC23(ITEM)                                               RSTART
      RC23(ITEM) = RR                                                 RSTART
      GO TO 492                                                       RSTART
  423 IF (ITEM.GT.2687) GO TO 490                                     RSTART
      IF (ITYPE.NE.2)   GO TO 490                                     RSTART
      IOLD = IC23(ITEM-1922)                                          RSTART
      IC23(ITEM-1922) = II                                            RSTART
      GO TO 494                                                       RSTART
C     COMMON /WINDFR/                                                 RSTART
   24 IF (ITEM.GT.150 ) GO TO 424                                     WINDOP
      IF (ITYPE.NE.1)   GO TO 490                                     RSTART
      ROLD = RC24(ITEM)                                               RSTART
      RC24(ITEM) = RR                                                 RSTART
      GO TO 492                                                       RSTART
  424 IF (ITEM.GT.1301) GO TO 490                                     WINDOP
      IF (ITYPE.NE.2)   GO TO 490                                     RSTART
      IOLD = IC24(ITEM-150)                                           WINDOP
```

```
      IC24(ITEM-150)= II                                               WINDOP
      GO TO 494                                                        RSTART
C                                                                      RSTART
C     ERROR MESSAGE - TERMINATE PROGRAM.                               RSTART
C                                                                      RSTART
  490 WRITE (6,491) AVAR,INDEX,NCOM,ITEM,ITYPE,RR,II,AA                RSTART
  491 FORMAT('0 SUBROUTINE RSTART INPUT ERROR'//                       RSTART
     *       ' AVAR= ',A8,'INDEX=',3I6,' NCOM=',I6,' ITEM=',I6,        RSTART
     * ' ITYPE=',I6,' RR=',G15.8,' II=',I8,' AA= ',A8//                RSTART
     * ' PROGRAM IS BEING TERMINATED.')                                RSTART
      STOP 2                                                           RSTART
C                                                                      RSTART
C     PRINT MESSAGE FOR REAL VARIABLES.                                RSTART
C                                                                      RSTART
  492 WRITE (6,493) AVAR,INDEX,COMMON(NCOM),ROLD,RR                    RSTART
  493 FORMAT('0',A6,'(',I4,',',I4,',',I4,') OF COMMON/',A6,'/',        RSTART
     *       ' HAS BEEN CHANGED FROM ',G15.8,' TO ',G15.8)             RSTART
      IF (RROLD.EQ.0.0) GO TO 400                                      RSTART
      IF (DABS(RROLD-ROLD).LE.0.00001*RROLD) GO TO 400                 RSTART
      WRITE (6,383) RROLD                                              RSTART
  383 FORMAT(' INPUT VALUE FOR RROLD WAS ',G15.8//)                    RSTART
      GO TO 490                                                        RSTART
C                                                                      RSTART
C     PRINT MESSAGE FOR INTEGER VARIABLES.                             RSTART
C                                                                      RSTART
  494 WRITE (6,495) AVAR,INDEX,COMMON(NCOM),IOLD,II                    RSTART
  495 FORMAT('0',A6,'(',I4,',',I4,',',I4,') OF COMMON/',A6,'/',        RSTART
     *       ' HAS BEEN CHANGED FROM ',    I8,' TO ',    I8)           RSTART
      IF (IIOLD.EQ.0) GO TO 400                                        RSTART
      IF (IOLD.EQ.IIOLD) GO TO 400                                     RSTART
      WRITE (6,385) IIOLD                                              RSTART
  385 FORMAT(' INPUT VALUE FOR IIOLD WAS ',I8//)                       RSTART
      GO TO 490                                                        RSTART
C                                                                      RSTART
C     PRINT MESSAGE FOR ALPHANUMERIC VARIABLES.                         RSTART
C                                                                      RSTART
  496 WRITE (6,497) AVAR,INDEX,COMMON(NCOM),AOLD,AA                    RSTART
  497 FORMAT('0',A6,'(',I4,',',I4,',',I4,') OF COMMON/',A6,'/',        RSTART
     *       ' HAS BEEN CHANGED FROM ',    A8,' TO ',    A8)           RSTART
      IF (AAOLD.EQ.BLANK) GO TO 400                                    RSTART
      AAOLD4 = AAOLD                                                   RSTART
      AOLD4  = AOLD                                                    RSTART
      IF (AOLD4.EQ.AAOLD4) GO TO 400                                   RSTART
      WRITE (6,387) AAOLD                                              RSTART
  387 FORMAT(' INPUT VALUE FOR AAOLD WAS ',A8//)                       RSTART
      GO TO 490                                                        RSTART
  999 CALL ELTIME(2,25)                                                RSTART
      RETURN                                                           RSTART
      END                                                              RSTART
```

```
      SUBROUTINE SEARCH(AVAR,INDEX,NCOM,ITEM)                           SEARCH
C                                                 REV IV    07/24/86SLIP
C     CALLED BY SUBROUTINE RSTART TO COMPUTE NCOM & ITEM FROM AVAR &    SEARCH
C     INDEX. RETURNS NCOM=0 FOR ERROR AND NCOM=50 FOR BLANK.            SEARCH
C                                                                       SEARCH
      IMPLICIT REAL*8(A-H,O-Z)                                          SEARCH
      DIMENSION BVAR(264),KOUNT(25),NDIM(3,264),NJ(3),NK(3),INDEX(3)    SLIP
      DIMENSION C1 (  17) , NC1 (  51)                                  PAGE
      DIMENSION C2 (   4) , NC2 (  12)                                  SEARCH
      DIMENSION C3 (  10) , NC3 (  30)                                  SEARCH
      DIMENSION C4 (   9) , NC4 (  27)                                  SEARCH
      DIMENSION C5 (   9) , NC5 (  27)                                  SLIP
      DIMENSION C6 (  30) , NC6 (  90)                                  SEARCH
      DIMENSION C7 (  11) , NC7 (  33)                                  SEARCH
      DIMENSION C8 (  10) , NC8 (  30)                                  TWOPI
      DIMENSION C9 (  15) , NC9 (  45)                                  SEARCH
      DIMENSION C10(  11) , NC10(  33)                                  SEARCH
      DIMENSION C11(  10) , NC11(  30)                                  SEARCH
      DIMENSION C12(   4) , NC12(  12)                                  SEARCH
      DIMENSION C13(  16) , NC13(  48)                                  SEARCH
      DIMENSION C14(   7) , NC14(  21)                                  SEARCH
      DIMENSION C15(  13) , NC15(  39)                                  SEARCH
      DIMENSION C16(  15) , NC16(  45)                                  SEARCH
      DIMENSION C17(   5) , NC17(  15)                                  SEARCH
      DIMENSION C18(   7) , NC18(  21)                                  SEARCH
      DIMENSION C19(   5) , NC19(  15)                                  SEARCH
      DIMENSION C20(  23) , NC20(  69)                                  SEARCH
      DIMENSION C21(   8) , NC21(  24)                                  CHGIII
      DIMENSION C22(   4) , NC22(  12)                                  SEARCH
      DIMENSION C23(  12) , NC23(  36)                                  SEARCH
      DIMENSION C24(   9) , NC24(  27)                                  WINDOP
C                                                                       SEARCH
      EQUIVALENCE (C1 (1),BVAR(  1)) , (NC1 (1),NDIM(1,  1))            SEARCH
      EQUIVALENCE (C2 (1),BVAR( 18)) , (NC2 (1),NDIM(1, 18))            PAGE
      EQUIVALENCE (C3 (1),BVAR( 22)) , (NC3 (1),NDIM(1, 22))            PAGE
      EQUIVALENCE (C4 (1),BVAR( 32)) , (NC4 (1),NDIM(1, 32))            PAGE
      EQUIVALENCE (C5 (1),BVAR( 41)) , (NC5 (1),NDIM(1, 41))            PAGE
      EQUIVALENCE (C6 (1),BVAR( 50)) , (NC6 (1),NDIM(1, 50))            SLIP
      EQUIVALENCE (C7 (1),BVAR( 80)) , (NC7 (1),NDIM(1, 80))            SLIP
      EQUIVALENCE (C8 (1),BVAR( 91)) , (NC8 (1),NDIM(1, 91))            SLIP
      EQUIVALENCE (C9 (1),BVAR(101)) , (NC9 (1),NDIM(1,101))            SLIP
      EQUIVALENCE (C10(1),BVAR(116)) , (NC10(1),NDIM(1,116))            SLIP
      EQUIVALENCE (C11(1),BVAR(127)) , (NC11(1),NDIM(1,127))            SLIP
      EQUIVALENCE (C12(1),BVAR(137)) , (NC12(1),NDIM(1,137))            SLIP
      EQUIVALENCE (C13(1),BVAR(141)) , (NC13(1),NDIM(1,141))            SLIP
      EQUIVALENCE (C14(1),BVAR(157)) , (NC14(1),NDIM(1,157))            SLIP
      EQUIVALENCE (C15(1),BVAR(164)) , (NC15(1),NDIM(1,164))            SLIP
      EQUIVALENCE (C16(1),BVAR(177)) , (NC16(1),NDIM(1,177))            SLIP
      EQUIVALENCE (C17(1),BVAR(192)) , (NC17(1),NDIM(1,192))            SLIP
      EQUIVALENCE (C18(1),BVAR(197)) , (NC18(1),NDIM(1,197))            SLIP
```

```
      EQUIVALENCE (C19(1),BVAR(204)) , (NC19(1),NDIM(1,204))           SLIP
      EQUIVALENCE (C20(1),BVAR(209)) , (NC20(1),NDIM(1,209))           SLIP
      EQUIVALENCE (C21(1),BVAR(232)) , (NC21(1),NDIM(1,232))           SLIP
      EQUIVALENCE (C22(1),BVAR(240)) , (NC22(1),NDIM(1,240))           SLIP
      EQUIVALENCE (C23(1),BVAR(244)) , (NC23(1),NDIM(1,244))           SLIP
      EQUIVALENCE (C24(1),BVAR(256)) , (NC24(1),NDIM(1,256))           SLIP
C                                                                      SEARCH
      DATA  NVAR/264/ , KOM/24/ , BLANK/8H        /                    SLIP
      DATA KOUNT/1,18,22,32,41,50,80,91,101,116,127,137,141,157,       SLIP
     *             164,177,192,197,204,209,232,240,244,256,265/        SLIP
C                                                                      SEARCH
C  1  COMMON/CONTRL/                                                   SEARCH
C                                                                      SEARCH
      DATA C1 / 8HTIME    ,8HNSEG    ,8HNJNT    ,8HNPL     ,8HNBLT    , SEARCH
     *          8HNBAG    ,8HNVEH    ,8HNGRND   ,8HNS      ,8HNQ      , SEARCH
     *          8HNSD     ,8HNFLX    ,8HNNRNSS  ,8HNWINDF  ,8HNJNTF   , SEARCH
     *          8HNPRT    ,8HNPG     /                                   PAGE
      DATA NC1 /  0,0,0   ,  0,0,0   ,  0,0,0   ,  0,0,0   ,  0,0,0   , SEARCH
     *            0,0,0   ,  0,0,0   ,  0,0,0   ,  0,0,0   ,  0,0,0   , SEARCH
     *            0,0,0   ,  0,0,0   ,  0,0,0   ,  0,0,0   ,  0,0,0   , SEARCH
     *            36,0,0  ,  0,0,0   /                                   PAGE
C                                                                      SEARCH
C  2  COMMON/CNTSRF/                                                   SEARCH
C                                                                      SEARCH
      DATA C2 / 8HPL      ,8HBELT    ,8HTPTS    ,8HBD      /           SEARCH
      DATA NC2 /  24,30,0 ,  20,8,0  ,  6,8,0   ,  24,40,0 /           EDGE
C                                                                      SEARCH
C  3  COMMON/VPOSTN/                                                   SEARCH
C                                                                      SEARCH
      DATA C3 / 8HZPLT    ,8HSPLT    ,8HAXV     ,8HVATAB   ,8HVT0     , SEARCH
     *          8HVDT     ,8HTIMEV   ,8HOMEGV   ,8HNVTAB   ,8HINDXV   / SEARCH
      DATA NC3 /  3,0,0   ,  3,0,0   ,  3,6,0   ,  6,501,6 ,  6,0,0   , VEHICL
     *            6,0,0   ,  6,0,0   ,  6,0,0   ,  6,0,0   ,  6,0,0   / SEARCH
C                                                                      SEARCH
C  4  COMMON/SGMNTS/                                                   SEARCH
C                                                                      SEARCH
      DATA C4 / 8HD       ,8HWMEG    ,8HWMEGD   ,8HU1      ,8HU2      , SEARCH
     *          8HSEGLP   ,8HSEGLV   ,8HSEGLA   ,8HNSYM    /           SEARCH
      DATA NC4 /  3,3,30  ,  3,30,0  ,  3,30,0  ,  3,30,0  ,  3,30,0  , SEARCH
     *            3,30,0  ,  3,30,0  ,  3,30,0  ,  30,0,0  /           SEARCH
C                                                                      SEARCH
C  5  COMMON/CMATRX/                                                   SEARCH
C                                                                      SEARCH
      DATA C5 / 8HV1      ,8HV2      ,8HV3      ,8HB12     ,8HA22     , SEARCH
     *          8HF       ,8HTQ      ,8HWJ      ,8HA11     /           SLIP
      DATA NC5 /  3,30,0  ,  3,30,0  ,  3,12,0  ,  3,3,60  ,  3,3,60  , SEARCH
     *            3,30,0  ,  3,30,0  ,  30,0,0  ,  3,3,60  /           SLIP
C                                                                      SEARCH
C  6  COMMON/ABDATA/                                                   SEARCH
C                                                                      SEARCH
```

```
      DATA C6 / 8HZDEP     ,8HDBR     ,8HDPVCTR  ,8HDEPLOY  ,8HAB      , SEARCH
     *          8HB        ,8HZR      ,8HBFB     ,8HDRR     ,8HVBAGG   , SEARCH
     *          8HVSCS     ,8HSPRK    ,8HCK      ,8HCMASS   ,8HCYMIN   , SEARCH
     *          8HCYMOUT   ,8HBAGPV   ,8HPD      ,8HVBAG    ,8HVOLBP   , SEARCH
     *          8HPCYV     ,8HPCYMIN  ,8HPVBAG   ,8HTV1     ,8HTV2     , SEARCH
     *          8HSWITCH   ,8HPYMOUT  ,8HSCALE   ,8HPREVT   ,8HIFULL   / SEARCH
      DATA NC6 /  3,5,0    ,  3,3,5   ,  3,5,0   ,  3,5,0   ,  3,5,0   , SEARCH
     *            9,4,5    ,  3,4,5   ,  3,4,5   ,  9,4,5   ,  5,0,0   , SEARCH
     *            5,0,0    ,  5,0,0   ,  5,0,0   ,  5,0,0   ,  5,0,0   , SEARCH
     *            5,0,0    ,  5,0,0   ,  5,0,0   ,  5,0,0   ,  5,0,0   , SEARCH
     *            5,0,0    ,  5,0,0   ,  5,0,0   ,  3,4,5   ,  3,10,5  , SEARCH
     *            5,0,0    ,  5,0,0   ,  5,0,0   ,  0,0,0   ,  6,0,0   / SEARCH
C                                                                        SEARCH
C   7 COMMON/TITLES/                                                     SEARCH
C                                                                        SEARCH
      DATA C7 / 8HDATE     ,8HCOMENT  ,8HVPSTTL  ,8HBDYTTL  ,8HBLTTTL  , SEARCH
     *          8HPLTTL    ,8HBAGTTL  ,8HSEG     ,8HJOINT   ,8HCGS     , SEARCH
     *          8HJS       /                                             SEARCH
      DATA NC7 /  3,0,0    ,  40,0,0  ,  20,0,0  ,  5,0,0   ,  5,8,0   , SEARCH
     *            5,30,0   ,  5,6,0   ,  30,0,0  ,  30,0,0  ,  30,0,0  , SEARCH
     *            30,0,0   /                                             SEARCH
C                                                                        SEARCH
C   8 COMMON/CNSNTS/                                                     SEARCH
C                                                                        SEARCH
      DATA C8 / 8HPI       ,8HRADIAN  ,8HG       ,8HTHIRD   ,8HEPS     , SEARCH
     *          8HUNITL    ,8HUNITM   ,8HUNITT   ,8HGRAVTY  ,8HTWOPI   / TWOPI
      DATA NC8 /  0,0,0    ,  0,0,0   ,  0,0,0   ,  0,0,0   ,  24,0,0  , SEARCH
     *            0,0,0    ,  0,0,0   ,  0,0,0   ,  3,0,0   ,   0,0,0  / TWOPI
C                                                                        SEARCH
C   9 COMMON/DESCRP/                                                     SEARCH
C                                                                        SEARCH
      DATA C9 / 8HPHI      ,8HW       ,8HRW      ,8HSR      ,8HHA      , SEARCH
     *          8HHB       ,8HRPHI    ,8HHT      ,8HSPRING  ,8HVISC    , SEARCH
     *          8HJNT      ,8HJPIN    ,8HJSING   ,8HJGLOB   ,8HJOINTF  / SEARCH
      DATA NC9 /  3,30,0   ,  30,0,0  ,  30,0,0  ,  4,60,0  ,  3,60,0  , SLIP
     *            3,60,0   ,  3,30,0  ,  3,3,60  ,  5,90,0  ,  7,90,0  , SEARCH
     *            30,0,0   ,  30,0,0  ,  30,0,0  ,  30,0,0  ,  30,0,0  / SEARCH
C                                                                        SEARCH
C  10 COMMON/JBARTZ/                                                     SEARCH
C                                                                        SEARCH
      DATA C10/ 8HMNPL     ,8HMNBLT   ,8HMNSEG   ,8HMNBAG   ,8HMPL     , SEARCH
     *          8HMBLT     ,8HMSEG    ,8HMBAG    ,8HNTPL    ,8HNTBLT   , SEARCH
     *          8HNTSEG    /                                             SEARCH
      DATA NC10/  30,0,0   ,  8,0,0   ,  30,0,0  ,  6,0,0   ,  3,5,30  , SEARCH
     *            3,5,8    ,  3,5,30  ,  3,10,6  ,  5,30,0  ,  5,8,0   , SEARCH
     *            5,30,0   /                                             SEARCH
C                                                                        SEARCH
C  11 COMMON/FORCES/                                                     SEARCH
C                                                                        SEARCH
      DATA C11/ 8HPSF      ,8HBSF     ,8HSSF     ,8HBAGSF   ,8HPRJNT   , SEARCH
```

```
*          8HNPANEL  ,8HNPSF    ,8HNBSF    ,8HNSSF    ,8HNBGSF    / SEARCH
      DATA NC11/ 7,70,0  ,  4,20,0  ,  10,40,0 ,  3,20,0  ,  7,30,0  ,  NCFORC
*           5,0,0    ,  0,0,0   ,  0,0,0   ,  0,0,0   ,  0,0,0   / SEARCH
C                                                                   SEARCH
C 12  COMMON/INTEST/                                                SEARCH
C                                                                   SEARCH
      DATA C12/ 8HSGTEST  ,8HXTEST   ,8HSEGT    ,8HREGT    /        SEARCH
      DATA NC12/  3,4,30  ,  3,120,0 ,  120,0,0 ,  120,0,0 /        SEARCH
C                                                                   SEARCH
C 13  COMMON/CSTRNT/                                                SEARCH
C                                                                   SEARCH
      DATA C13/ 8HA13     ,8HA23     ,8HB31     ,8HB32     ,8HHHT     , SEARCH
*          8HRK1     ,8HRK2     ,8HQQ      ,8HTQQ     ,8HRQQ     , SEARCH
*          8HHQQ     ,8HSQQ     ,8HCFQQ    ,8HKQ1     ,8HKQ2     , SEARCH
*          8HKQTYPE  /                                              SEARCH
      DATA NC13/  3,3,24  ,  3,3,24  ,  3,3,24  ,  3,3,24  ,  3,3,12  , SEARCH
*           3,12,0   ,  3,12,0  ,  3,12,0  ,  3,12,0  ,  3,12,0  , SEARCH
*           3,12,0   ,  12,0,0  ,  12,0,0  ,  12,0,0  ,  12,0,0  , SEARCH
*           12,0,0   /                                              SEARCH
C                                                                   SEARCH
C 14  COMMON/TABLES/                                                SEARCH
C                                                                   SEARCH
      DATA C14/ 8HMXNTI   ,8HMXNTB   ,8HMXTB1   ,8HMXTB2   ,8HNTI     , SEARCH
*          8HNTAB    ,8HTAB     /                                   SEARCH
      DATA NC14/  0,0,0   ,  0,0,0   ,  0,0,0   ,  0,0,0   ,  50,0,0  , SEARCH
*           1250,0,0 , 4500,0,0/                                    BUTLER2
C                                                                   SEARCH
C 15  COMMON/COMAIN/                                                SEARCH
C                                                                   SEARCH
      DATA C15/ 8HVAR     ,8HDER     ,8HDT      ,8HHO      ,8HHMAX    , SEARCH
*          8HHMIN    ,8HRSTIME  ,8HISTEP   ,8HNSTEPS  ,8HNDINT   , SEARCH
*          8HNEQ     ,8HIRSIN   ,8HIRSOUT  /                        SEARCH
      DATA NC15/  240,0,0 ,  240,0,0 ,  0,0,0   ,  0,0,0   ,  0,0,0   , SEARCH
*           0,0,0    ,  0,0,0   ,  0,0,0   ,  0,0,0   ,  0,0,0   , SEARCH
*           0,0,0    ,  0,0,0   ,  0,0,0   /                        SEARCH
C                                                                   SEARCH
C 16  COMMON/CDINT /                                                SEARCH
C                                                                   SEARCH
      DATA C16/ 8HUU      ,8HGH      ,8HE       ,8HFF      ,8HGG      , SEARCH
*          8HY       ,8HU       ,8HH       ,8HHPRINT  ,8HHS      , SEARCH
*          8HTPRINT  ,8HTSTART  ,8HICNT    ,8HIDBL    ,8HIFLAG   / SEARCH
      DATA NC16/  4,0,0   ,  3,4,0   ,  3,240,0 ,  5,240,0 ,  5,240,0 , SEARCH
*           5,240,0  ,  5,240,0 ,  0,0,0   ,  0,0,0   ,  0,0,0   , SEARCH
*           0,0,0    ,  0,0,0   ,  0,0,0   ,  0,0,0   ,  0,0,0   / SEARCH
C                                                                   SEARCH
C 17  COMMON/DAMPER/                                                SEARCH
C                                                                   SEARCH
      DATA C17/ 8HAPSDM   ,8HAPSDN   ,8HASD     ,8HMSDM    ,8HMSDN    / SEARCH
      DATA NC17/  3,20,0  ,  3,20,0  ,  5,20,0  ,  20,0,0  ,  20,0,0  / SEARCH
C                                                                   SEARCH
```

```
C 18  COMMON/CEULER/                                                        SEARCH
C                                                                           SEARCH
      DATA C18/ 8HIEULER  ,8HHIR     ,8HANG     ,8HANGD    ,8HFE      , SEARCH
     *          8HTQE     ,8HCONST   /                                    SEARCH
      DATA NC18/  30,0,0  ,  3,3,90  ,  3,30,0  ,  3,30,0  ,  3,30,0  , JDRIFT
     *            3,30,0  ,  5,30,0  /                                    JDRIFT
C                                                                           SEARCH
C 19  COMMON/TEMPVI/                                                        SEARCH
C                                                                           SEARCH
      DATA C19/ 8HCREST   ,8HTTI     ,8HR1I     ,8HR2I     ,8HJSTOP   / SEARCH
      DATA NC19/  0,0,0   ,  3,0,0   ,  3,0,0   ,  3,0,0   ,  4,2,30  / SEARCH
C                                                                           SEARCH
C 20  COMMON/CYDATA/                                                        SEARCH
C                                                                           SEARCH
      DATA C20/ 8HCYTD    ,8HCYPA    ,8HCYSP    ,8HCYTO    ,8HCYVO    , SEARCH
     *          8HCYCD    ,8HCYK     ,8HCYR     ,8HCYAT    ,8HCYPV    , SEARCH
     *          8HCYCDO   ,8HCYAO    ,8HCYPO    ,8HCYSS    ,8HCYLO    , SEARCH
     *          8HCYC     ,8HCYRHOO  ,8HCYVMAX  ,8HCYORFC  ,8HCYRHO   , SEARCH
     *          8HCYT     ,8HCYRHP   ,8HCYV     /                         SEARCH
      DATA NC20/  5,0,0   ,  5,0,0   ,  5,0,0   ,  5,0,0   ,  5,0,0   , SEARCH
     *            5,0,0   ,  5,0,0   ,  5,0,0   ,  5,0,0   ,  5,0,0   , SEARCH
     *            5,0,0   ,  5,0,0   ,  5,0,0   ,  5,0,0   ,  5,0,0   , SEARCH
     *            5,0,0   ,  5,0,0   ,  5,0,0   ,  5,0,0   ,  5,0,0   , SEARCH
     *            5,0,0   ,  5,0,0   ,  5,0,0   /                         SEARCH
C                                                                           SEARCH
C 21  COMMON/RSAVE/                                                         SEARCH
C                                                                           SEARCH
      DATA C21/ 8HXSG     ,8HDPMI    ,8HLPMI    ,8HNSG     ,8HMSG     , SEARCH
     *          8HMCG     ,8HMCGIN   ,8HKREF    /                         CHGIII
      DATA NC21/  3,20,3  ,  3,3,30  ,  30,0,0  ,  9,0,0   ,  20,9,0  , WINDOP
     *            1,0,0   ,  24,5,0  ,  20,9,0  /                         TTHKREF
C                                                                           SEARCH
C 22  COMMON/FLXBLE/                                                        SEARCH
C                                                                           SEARCH
      DATA C22/ 8HHF      ,8HB42     ,8HV4      ,8HNFLEX   /              SEARCH
      DATA NC22/  4,12,8  ,  3,3,24  ,  3,8,0   ,  3,8,0   /              SEARCH
C                                                                           SEARCH
C 23  COMMON/HRNESS/                                                        SEARCH
C                                                                           SEARCH
      DATA C23/ 8HBAR     ,8HBB      ,8HBBDOT   ,8HPLOSS   ,8HXLONG   , SEARCH
     *          8HHTIME   ,8HIBAR    ,8HNL      ,8HNPTSPB  ,8HNPTPLY  , SEARCH
     *          8HNTHRNS  ,8HNBLTPH  /                                    SEARCH
      DATA NC23/  15,100,0,  100,0,0 ,  100,0,0 ,  2,100,0 ,  20,0,0  , SEARCH
     *            2,0,0   ,  5,100,0 ,  2,100,0 ,  20,0,0  ,  20,0,0  , SEARCH
     *            20,0,0  ,  5,0,0   /                                    SEARCH
C                                                                           SEARCH
C 24  COMMON/WINDFR/                                                        SEARCH
      DATA C24/ 8HWTIME   ,8HQFU     ,8HQFV     ,8HWF      ,8HIWIND   , WINDOP
     *          8HMWSEG   ,8HNFVSEG  ,8HNFVNT   ,8HMOWSEG  /              WINDOP
      DATA NC24/  30,0,0  ,  3, 5,0  ,  3, 5,0  ,  3,30,0  ,  30,0,0  , WINDOP
```

```
     *              7,30,0   ,  6,0,0    ,  5,0,0    ,  30,30,0 /                WINDOP
      NCOM = 50                                                                  SEARCH
      IF (AVAR.EQ.BLANK) GO TO 99                                                SEARCH
C                                                                                SEARCH
C     SEARCH FOR VARIABLE NO. IV.                                                SEARCH
C                                                                                SEARCH
      NCOM = 0                                                                   SEARCH
      DO 10 IV=1,NVAR                                                            SEARCH
      IF (AVAR.EQ.BVAR(IV)) GO TO 12                                             SEARCH
   10 CONTINUE                                                                   SEARCH
      GO TO 99                                                                   SEARCH
C                                                                                SEARCH
C     SEARCH FOR COMMON NO. IC.                                                  SEARCH
C                                                                                SEARCH
   12 DO 20 IC=1,KOM                                                             SEARCH
      IF (IV.GE.KOUNT(IC).AND.IV.LT.KOUNT(IC+1)) GO TO 22                        SEARCH
   20 CONTINUE                                                                   SEARCH
      GO TO 99                                                                   SEARCH
C                                                                                SEARCH
C     COMPUTE ITEM NO. FOR VARIABLE IV IN COMMON IC.                             SEARCH
C                                                                                SEARCH
   22 K1 = KOUNT(IC)                                                             SEARCH
      K2 = IV-1                                                                  SEARCH
      ITEM = 1                                                                   SEARCH
      IF (K1.EQ.IV) GO TO 25                                                     SEARCH
      DO 24 K=K1,K2                                                              SEARCH
      NI = 1                                                                     SEARCH
      DO 23 I=1,3                                                                SEARCH
      IF (NDIM(I,K).NE.0) NI=NI*NDIM(I,K)                                        SEARCH
   23 CONTINUE                                                                   SEARCH
   24 ITEM = ITEM+NI                                                             SEARCH
   25 DO 26 I=1,3                                                                SEARCH
      IF (INDEX(I).EQ.0 .AND. NDIM(I,IV).NE.0) GO TO 99                          SEARCH
      IF (NDIM(I,IV).EQ.0 .AND. INDEX(I).GT.1) GO TO 99                          SEARCH
      NJ(I) = MAX0(INDEX(I)-1,0)                                                 SEARCH
      NK(I) = MAX0(NDIM(I,IV),1)                                                 SEARCH
      IF (NJ(I).GE.NK(I)) GO TO 99                                               SEARCH
   26 CONTINUE                                                                   SEARCH
      ITEM = ITEM+NJ(1)+NJ(2)*NK(1)+NJ(3)*NK(2)*NK(1)                            SEARCH
      NCOM = IC                                                                  SEARCH
   99 RETURN                                                                     SEARCH
      END                                                                        SEARCH
```

```
      SUBROUTINE SEGSEG(M,MM,N,NS,NT)                                   SEGSEG
C                                                  REV IV    02/07/87HYPER
      IMPLICIT REAL*8(A-H,O-Z)                                          SEGSEG
      COMMON/TABLES/MXNTI,MXNTB,MXTB1,MXTB2,NTI(50),NTAB(1250),TAB(4500)SEGSEG
      COMMON/CNTSRF/ PL(24,30),BELT(20,8),TPTS(6,8),BD(24,40)           EDGE
      COMMON/SGMNTS/ D(3,3,30),WMEG(3,30),WMEGD(3,30),U1(3,30),U2(3,30),SEGSEG
     *               SEGLP(3,30),SEGLV(3,30),SEGLA(3,30),NSYM(30)       SEGSEG
      COMMON/FORCES/PSF(7,70),BSF(4,20),SSF(10,40),BAGSF(3,20),         NCFORC
     *               PRJNT(7,30),NPANEL(5),NPSF,NBSF,NSSF,NBGSF         SEGSEG
      COMMON/CSTRNT/ A13(3,3,24),A23(3,3,24),B31(3,3,24),B32(3,3,24),   SEGSEG
     *               HHT(3,3,12),RK1(3,12),RK2(3,12),QQ(3,12),TQQ(3,12),SEGSEG
     *               RQQ(3,12),HQQ(3,12),SQQ(12),CFQQ(12),              SEGSEG
     *               KQ1(12),KQ2(12),KQTYPE(12)                         SEGSEG
      COMMON/RSAVE/  XSG(3,20,3),DPMI(3,3,30),LPMI(30),                 TGMOD7
     *               NSG(9),MSG(20,9),MCG,MCGIN(24,5),KREF(20,9)        TGMOD7
      COMMON/TEMPVS/DMNT(3,3),TEMP(3,3),B(3,3),XMN(3),RLN(3),XMM(3),    SEGSEG
     *            TM(3),R(3),RM(3),DMNWN(3),RLM(3),RN(3),VMN(3),VR(3),  SEGSEG
     *            WNM(3),WCM(3),WCN(3),VREL(3),FFM(3),FR(3),TQM(3),     SEGSEG
     *            TQN(3),TQNT(3),T(3),H(3),T1(3),T2(3),RMD(3),RND(3),   SEGSEG
     *            TD(3),TT4(3,4),TT5(3,4),T3(3),T4(3),P,AMR,FM,CF,      SEGSEG
     *            VRM,VRT,VRTS,VRTEST,TF,ELOSS,MCF,NCF,T5(3),T6(3)      TGMOD7
      CALL ELTIME(1,23)                                                 SEGSEG
C                                                                       EDGE
C     COMPUTATIONS ARE DONE IN M'S REFERENCE SYSTEM                     EDGE
      NN = IABS(NS)                                                     SEGSEG
      CALL DOTT33(D(1,1,M),D(1,1,N),DMNT)                               SEGSEG
      DO 10 I = 1,3                                                     SEGSEG
   10 XMN(I) = SEGLP(I,M) - SEGLP(I,N)                                  SEGSEG
      CALL MAT31(D(1,1,M),XMN,XMM)                                      SEGSEG
      J = 3                                                             HYPER
      IF(BD(1,NN).LT.0.0)J = 4                                          HYPER
      CALL MAT31(DMNT,BD(J+1,NN),RLN)                                   HYPER
      J = 3                                                             HYPER
      IF(BD(1,MM).LT.0.0)J = 4                                          HYPER
      DO 15 I = 1,3                                                     EDGE
      J = J + 1                                                         HYPER
   15 R(I) = RLN(I) - XMM(I) - BD(J,MM)                                 HYPER
      LT = NTAB(NT)                                                     SEGSEG
      TB = 1.0                                                          EDGE
      IF((BD(1,MM).GT.0.0).AND.(BD(1,NN).GT.0.0))GO TO 20               HYPER
C NEW HYPERELLIPSOID - AT LEAST ONE SURFACE IS A HYPERELLIPSOID         HYPER
      IF (BD(1,MM).LT.0.0.AND.BD(23,MM).NE.0.0) STOP 23                 HYPER
      IF (BD(1,NN).LT.0.0.AND.BD(23,NN).NE.0.0) STOP 23                 HYPER
C A HYPERELLIPSOID MUST HAVE IDENTICAL POWERS.                          HYPER
C IF(NS.LT.0)    STOP - INTERIOR INTERSECTION NOT OPERATIONAL           HYPER
      IF(NS.LT.0) STOP 38                                               HYPER
      IF(TAB(LT+23).LE.1.0) CALL HYEST(BD(1,MM),BD(1,NN),TAB(LT+22))    HYPER
      IF(TAB(LT+23).GT.1.0) CALL HYNTR(BD(1,MM),BD(1,NN),TAB(LT+22))    HYPER
      BET = TAB(LT+23)                                                  HYPER
      IF(BET.GT.1.0)TB = 1.0/BET                                        HYPER
```

```
      GO TO 25                                                          HYPER
C OLD ELLIPSOIDS                                                        HYPER
   20 IF(NS.LT.0.0)TB = -TB                                             HYPER
      CALL DOTT33(BD(7,NN),DMNT,TEMP)                                   EDGE
      CALL MAT33(DMNT,TEMP,B)                                           EDGE
      CALL INTERS(BD(7,MM),B,R,TB,RM,TAB(LT+22),TM)                     SEGSEG
C                          A    B R    Z      C      AZ                 EDGE
C     INTERS SOLVES (CA + B)Z = BR,   TB = SQRT(Z.AZ)                   EDGE
C                                                                       EDGE
   25 MCF = NTAB(NT+1)                                                  HYPER
      NCF = -MCF                                                        SEGSEG
      IF(NCF.GT.0)CFQQ(NCF) = -999.                                     SEGSEG
C                                                                       EDGE
C     CHECK FOR INTERSECTION                                            EDGE
C                                                                       EDGE
      IF(TB.GE.1.0)GO TO 75                                             HYPER
      S1 = 0.0                                                          SEGSEG
      S2 = 0.0                                                          SEGSEG
      DO 30 I = 1,3                                                     HYPER
      RI = R(I)                                                         SEGSEG
      IF(NS.LT.0)RI = RM(I) + TB*(RM(I) - R(I))                         SEGSEG
      S1 = S1 + RI**2                                                   SEGSEG
   30 S2 = S2 + TM(I)**2                                                HYPER
      AMR = DSQRT(S2)                                                   SEGSEG
      P = (1.0/TB - 1.0)*DSQRT(S1)                                      SEGSEG
      J = 3                                                             HYPER
      IF(BD(1,MM).LT.0.0)J = 4                                          HYPER
      DO 35 I = 1,3                                                     HYPER
      J = J + 1                                                         HYPER
      IF((BD(1,MM).LT.0.0).OR.(BD(1,NN).LT.0.0))RM(I) = TB*RM(I)        HYPER
      TM(I) = -TM(I)/AMR                                                SEGSEG
      T2(I) = RM(I) - R(I)                                              SEGSEG
      RN(I) = T2(I) + RLN(I)                                            SEGSEG
   35 RLM(I) = RM(I) + BD(J,MM)                                         HYPER
      CALL DOT31(DMNT,RN,RLN)                                           SEGSEG
      CALL PLSEGF(M,N,NT)                                               SEGSEG
C                                                                       EDGE
C     STORE PRINT DATA                                                  EDGE
C                                                                       EDGE
      SSF(1,NSSF) = P                                                   SEGSEG
      DO 40 I = 1,3                                                     HYPER
      SSF(I+4,NSSF) = RLM(I)                                            EDGE
   40 SSF(I+7,NSSF) = RLN(I)                                            HYPER
      IF(LPMI(M).NE.0) CALL DOT31(DPMI(1,1,M),RLM,SSF(5,NSSF))          EDGE
      IF(LPMI(N).NE.0) CALL DOT31(DPMI(1,1,N),RLN,SSF(8,NSSF))          EDGE
      IF(MCF.LT.0)GO TO 45                                              HYPER
      SSF(2,NSSF) = FM                                                  SEGSEG
      TF2FM2 = TF**2 - FM**2                                            SEGSEG
      IF(TF2FM2.LT.0.0)TF2FM2 = 0.0                                     SEGSEG
      SSF(3,NSSF) = DSQRT(TF2FM2)                                       SEGSEG
```

```
      SSF(4,NSSF) = TF                                                  SEGSEG
      GO TO 75                                                          HYPER
C                                                                       EDGE
C     ROLL-SLIDE                                                        EDGE
   45 DO 50 I = 1,3                                                     HYPER
   50 SSF(I+1,NSSF) = T(I)                                              HYPER
      IF((BD(1,MM).LT.0.0).OR.(BD(1,NN).LT.0.0)) STOP 29                HYPER
      ANR = XDY(TM,B,T2)                                                SEGSEG
      CALL CROSS(TM,WNM,T2)                                             SEGSEG
      CALL MAT31(B,VR,T1)                                               SEGSEG
      TB = TM(1)*T1(1) + TM(2)*T1(2) + TM(3)*T1(3)                      EDGE
      DO 60 I = 1,3                                                     HYPER
      DO 55 J = 1,3                                                     HYPER
      K = I + 3*(J+1)                                                   SEGSEG
      TT4(I,J) = BD(K,MM)/AMR + B(I,J)/ANR                              SEGSEG
   55 TT5(I,J) = TT4(I,J)                                               HYPER
      TT4(I,4) = T2(I) - (T1(I) - TB*TM(I))/ANR                         EDGE
   60 TT5(I,4) = TM(I)                                                  HYPER
      CALL DSMSOL(TT4,3,3)                                              SEGSEG
      CALL DSMSOL(TT5,3,3)                                              SEGSEG
      S1 =  TM(1)*TT4(1,4) + TM(2)*TT4(2,4) + TM(3)*TT4(3,4)            EDGE
      S2 = (TM(1)*TT5(1,4) + TM(2)*TT5(2,4) + TM(3)*TT5(3,4))/S1        EDGE
      DO 65 I = 1,3                                                     HYPER
      RMD(I) = TT4(I,4) - S2*TT5(I,4)                                   EDGE
   65 RND(I) = RND(I) + VR(I)                                           HYPER
      CALL CROSS(DMNWN,RND,T1)                                          EDGE
      CALL CROSS(WMEG(1,MM),RMD,T2)                                     EDGE
      CALL MAT31(B,RND,T3)                                              EDGE
      CALL CROSS(DMNWN,TM,T4)                                           EDGE
      S1 = TM(1)*T3(1) + TM(2)*T3(2) + TM(3)*T3(3)                      EDGE
      SQQ(NCF) = 0.0                                                    SEGSEG
      DO 70 I = 1,3                                                     HYPER
      T1(I) = T1(I) - T2(I)                                             EDGE
   70 SQQ(NCF)=SQQ(NCF)+TM(I)*T1(I)-VR(I)*(T4(I)+(T3(I)-S1*TM(I))/ANR)  HYPER
      CALL DOT31(D(1,1,M),T1,RQQ(1,NCF))                                EDGE
   75 CALL ELTIME(2,23)                                                 HYPER
      RETURN                                                            SEGSEG
      END                                                               SEGSEG
```

```
      SUBROUTINE SETUP1                                                 SETUP1
C                                                   REV IV    07/24/86SLIP
C     FOR KK=1 (BEFORE CONTACT ROUTINE IN DAUX)                         SETUP1
C     SET UP INITIAL VALUES OF A2 AND B2 ARRAYS FOR THIS TIME POINT.    SETUP1
C     SET UP INITIAL VALUES OF ARRAYS U1,U2 AND V1.                     SETUP1
C                                                                       SETUP1
      IMPLICIT REAL*8(A-H,O-Z)                                          SETUP1
      COMMON/CONTRL/ TIME,NSEG,NJNT,NPL,NBLT,NBAG,NVEH,NGRND,           SETUP1
     *               NS,NQ,NSD,NFLX,NHRNSS,NWINDF,NJNTF,NPRT(36),NPG    PAGE
      COMMON/SGMNTS/ D(3,3,30),WMEG(3,30),WMEGD(3,30),U1(3,30),U2(3,30),SETUP1
     *               SEGLP(3,30),SEGLV(3,30),SEGLA(3,30),NSYM(30)       SETUP1
      COMMON/DESCRP/ PHI(3,30),W(30),RW(30),SR(4,60),HA(3,60),HB(3,60), SLIP
     *               RPHI(3,30),HT(3,3,60),SPRING(5,90),VISC(7,90),     SETUP1
     *               JNT(30),IPIN(30),ISING(30),IGLOB(30),JOINTF(30)    SETUP1
      COMMON/CMATRX/ V1(3,30),V2(3,30),V3(3,12),B12(3,3,60),A22(3,3,60),SETUP1
     *               F(3,30),TQ(3,30),WJ(30),A11(3,3,30)                SLIP
      COMMON/CEULER/ IEULER(30),HIR(3,3,90),ANG(3,30),ANGD(3,30),       SLIP
     *               FE(3,30),TQE(3,30),CONST(5,30)                     SLIP
      COMMON/TEMPVS/T(3),S(3),T1(3),T2(3),T3(3),T4(3),T5(3),T6(3),      SETUP1
     *              T7(3),T8(3),T9(3),T10(3),T11(3),T12(3),HH(3),       SETUP1
     *              TT1(3,3),TT2(3,3),S1,SQS1,S2,S3,S4,V1T,SR2          SLIP
      DATA IFIRST/1/                                                    SLIP
C                                                                       SETUP1
      CALL ELTIME(1,10)                                                 SETUP1
      IF (IFIRST.EQ.0) GO TO 15                                         SLIP
      IF (NJNT.EQ.0) GO TO 15                                           SLIP
      DO 10 I = 1,NJNT                                                  SLIP
      DO 10 J = 1,3                                                     SLIP
      DO  8 K = 1,3                                                     SLIP
    8 A11(J,K,I) = 0.0                                                  SLIP
   10 A11(J,J,I) = 1.0                                                  SLIP
      IFIRST = 0                                                        SLIP
   15 DO 20 I=1,NGRND                                                   SLIP
C                                                                       SETUP1
C     SET EACH U1N = 0                                                  SETUP1
C                                                                       SETUP1
      U1(1,I) = 0.0                                                     SETUP1
      U1(2,I) = 0.0                                                     SETUP1
      U1(3,I) = 0.0                                                     SETUP1
C                                                                       SETUP1
C     SET EACH U2N = WNX(PHIN*WN)                                       SETUP1
C                                                                       SETUP1
      U2(1,I) = WMEG(2,I)*WMEG(3,I) * (PHI(2,I)-PHI(3,I))               SETUP1
      U2(2,I) = WMEG(1,I)*WMEG(3,I) * (PHI(3,I)-PHI(1,I))               SETUP1
   20 U2(3,I) = WMEG(1,I)*WMEG(2,I) * (PHI(1,I)-PHI(2,I))               SETUP1
      IF (NPRT(11).NE.0) WRITE (6,21) ((U2(I,J),I=1,3),J=1,NSEG)        SETUP1
   21 FORMAT(' U2 ARRAY'/(1X,1P9D14.4))                                 SETUP1
      IF (NJNT.LE.0) GO TO 98                                           SETUP1
      DO 40 J=1,NJNT                                                    SETUP1
      DO 31 K=1,3                                                       SETUP1
```

```
      T1(K) = SR(K,2*J-1)                                               SLIP
      T2(K) = SR(K,2*J  )                                               SLIP
      IF (IABS(IPIN(J)).LT.5) GO TO 31                                  SLIP
      IF (IEULER(J).EQ.-1) GO TO 31                                     SLIP
      T1(K) = T1(K) + SR(4,2*J-1)*HT(K,3,2*J-1)                         SLIP
   31 V1(K,J) = 0.0                                                     SETUP1
      I = IABS(JNT(J))                                                  SETUP1
      IF (I.LE.0) GO TO 40                                              SETUP1
C                                                                       SETUP1
C     FOR EACH JOINT SET                                                SETUP1
C     B12(2J-1) = B12(J,I  ) = -D(I)' * SR(2J-1) X                      SETUP1
C     B12(2J  ) = B12(J,J+1) =  D(J+1)' * SR(2J) X                      SETUP1
C                                                                       SETUP1
      B12(1,1,2*J-1) = D(3,1,I)*T1(2) - D(2,1,I)*T1(3)                  SLIP
      B12(2,1,2*J-1) = D(3,2,I)*T1(2) - D(2,2,I)*T1(3)                  SLIP
      B12(3,1,2*J-1) = D(3,3,I)*T1(2) - D(2,3,I)*T1(3)                  SLIP
      B12(1,2,2*J-1) = D(1,1,I)*T1(3) - D(3,1,I)*T1(1)                  SLIP
      B12(2,2,2*J-1) = D(1,2,I)*T1(3) - D(3,2,I)*T1(1)                  SLIP
      B12(3,2,2*J-1) = D(1,3,I)*T1(3) - D(3,3,I)*T1(1)                  SLIP
      B12(1,3,2*J-1) = D(2,1,I)*T1(1) - D(1,1,I)*T1(2)                  SLIP
      B12(2,3,2*J-1) = D(2,2,I)*T1(1) - D(1,2,I)*T1(2)                  SLIP
      B12(3,3,2*J-1) = D(2,3,I)*T1(1) - D(1,3,I)*T1(2)                  SLIP
C                                                                       SETUP1
      B12(1,1,2*J  ) = D(2,1,J+1)*T2(3) - D(3,1,J+1)*T2(2)              SLIP
      B12(2,1,2*J  ) = D(2,2,J+1)*T2(3) - D(3,2,J+1)*T2(2)              SLIP
      B12(3,1,2*J  ) = D(2,3,J+1)*T2(3) - D(3,3,J+1)*T2(2)              SLIP
      B12(1,2,2*J  ) = D(3,1,J+1)*T2(1) - D(1,1,J+1)*T2(3)              SLIP
      B12(2,2,2*J  ) = D(3,2,J+1)*T2(1) - D(1,2,J+1)*T2(3)              SLIP
      B12(3,2,2*J  ) = D(3,3,J+1)*T2(1) - D(1,3,J+1)*T2(3)              SLIP
      B12(1,3,2*J  ) = D(1,1,J+1)*T2(2) - D(2,1,J+1)*T2(1)              SLIP
      B12(2,3,2*J  ) = D(1,2,J+1)*T2(2) - D(2,2,J+1)*T2(1)              SLIP
      B12(3,3,2*J  ) = D(1,3,J+1)*T2(2) - D(2,3,J+1)*T2(1)              SLIP
C                                                                       SETUP1
C     NOTE THAT FOR EACH JOINT                                          SETUP1
C     A21(M,N) = B12(N,M)                                               SETUP1
C                                                                       SETUP1
C     FOR EACH JOINT SET                                                SETUP1
C     V1(J) = -D(I)'*W(I)X( W(I)XSR(2J-1) )                             SETUP1
C             +D(J+1)'*W(J+1)X( W(J+1)XSR(2J) )                         SETUP1
C                                                                       SETUP1
      CALL CROSS(WMEG(1,I),T1,T)                                        SLIP
      CALL CROSS(WMEG(1,I),T,S)                                         SETUP1
      CALL DOT31(D(1,1,I),S,V1(1,J))                                    SETUP1
      CALL CROSS(WMEG(1,J+1),T2,T)                                      SLIP
      CALL CROSS(WMEG(1,J+1),T,S)                                       SETUP1
      CALL DOT31(D(1,1,J+1),S,T)                                        SETUP1
      DO 32 K=1,3                                                       SLIP
   32 V1(K,J) = T(K) - V1(K,J)                                          SLIP
      IF (IABS(IPIN(J)).LT.5) GO TO 40                                  SLIP
      IF (IEULER(J).EQ.-1) GO TO 40                                     SLIP
```

```
      CALL DOT31(D(1,1,I),HT(1,3,2*J-1),T4)                             SLIP
      CALL CROSS(WMEG(1,I),HT(1,3,2*J-1),T5)                            SLIP
      CALL DOT31(D(1,1,I),T5,T6)                                        SLIP
      V1T = V1(1,J)*T4(1) + V1(2,J)*T4(2) + V1(3,J)*T4(3)               SLIP
      SR2 = 2.0*SR(4,2*J)                                               SLIP
      DO 34 K = 1,3                                                     SLIP
      V1(K,J) = V1(K,J) - V1T*T4(K) - SR2*T6(K)                         SLIP
      S1=T4(1)*B12(1,K,2*J-1)+T4(2)*B12(2,K,2*J-1)+T4(3)*B12(3,K,2*J-1) SLIP
      S2=T4(1)*B12(1,K,2*J  )+T4(2)*B12(2,K,2*J  )+T4(3)*B12(3,K,2*J  ) SLIP
      DO 33 L = 1,3                                                     SLIP
      A11(K,L,J) = -T4(K)*T4(L)                                         SLIP
      B12(L,K,2*J-1) = B12(L,K,2*J-1) - S1*T4(L)                        SLIP
   33 B12(L,K,2*J  ) = B12(L,K,2*J  ) - S2*T4(L)                        SLIP
   34 A11(K,K,J) = 1.0 + A11(K,K,J)                                     SLIP
   40 CONTINUE                                                          SETUP1
      IF (NPRT(11).NE.0) WRITE (6,41) ((V1(I,J),I=1,3),J=1,NJNT)        SETUP1
   41 FORMAT(' V1 ARRAY'/(1X,1P9D14.4))                                 SETUP1
C                                                                       SETUP1
C     IF IPIN(M)=1, SET V2(M)=(WN.HN-WM.HM)DN'WNXHN                     SETUP1
C                                                                       SETUP1
      DO 50 J=1,NJNT                                                    SETUP1
      DO 43 K=1,3                                                       SETUP1
   43 V2(K,J) = 0.0                                                     SETUP1
      IF (IPIN(J).LT.1) GO TO 50                                        SLIP
      IF (IPIN(J).GT.1.AND.IPIN(J).LT.6) GOTO 50                        SLIP
      I = IABS(JNT(J))                                                  SETUP1
      CALL CROSS (WMEG(1,I  ),HB(1,2*J-1),T)                            SETUP1
      CALL DOT31 (D(1,1,I  ),T,T1)                                      SETUP1
C     CALL CROSS (WMEG(1,J+1),HB(1,2*J  ),T)                            SETUP1
C     CALL DOT31 (D(1,1,J+1),T,T2)                                      SETUP1
      S1 = WMEG(1,I)*HB(1,2*J-1)                                        SETUP1
     *   + WMEG(2,I)*HB(2,2*J-1)                                        SETUP1
     *   + WMEG(3,I)*HB(3,2*J-1)                                        SETUP1
      S2 = WMEG(1,J+1)*HB(1,2*J)                                        SETUP1
     *   + WMEG(2,J+1)*HB(2,2*J)                                        SETUP1
     *   + WMEG(3,J+1)*HB(3,2*J)                                        SETUP1
      DO 44 K=1,3                                                       SETUP1
C  44 V2(K,J) = S1*T1(K) - S2*T2(K)                                     SETUP1
   44 V2(K,J) = (S1-S2)*T1(K)                                           SETUP1
   50 CONTINUE                                                          SETUP1
   98 CALL ELTIME(2,10)                                                 SETUP1
      RETURN                                                            SETUP1
      END                                                               SETUP1
```

```
      SUBROUTINE SETUP2                                                 SETUP2
C                                                    REV IV    07/24/86SLIP
C     CALLED BY DAUX AFTER CONTACT ROUTINES AND BY UPDATE PRIOR TO      SETUP2
C     DAUX TO SET UP A2 ARRAY AND (FOR NQ*0) THE A13,A23 AND V3 ARRAYS. SETUP2
C                                                                       SETUP2
      IMPLICIT REAL*8(A-H,O-Z)                                          SETUP2
      COMMON/CONTRL/ TIME,NSEG,NJNT,NPL,NBLT,NBAG,NVEH,NGRND,           SETUP2
     *               NS,NQ,NSD,NFLX,NHRNSS,NWINDF,NJNTF,NPRT(36),NPG    PAGE
      COMMON/SGMNTS/ D(3,3,30),WMEG(3,30),WMEGD(3,30),U1(3,30),U2(3,30),SETUP2
     *               SEGLP(3,30),SEGLV(3,30),SEGLA(3,30),NSYM(30)       SETUP2
      COMMON/DESCRP/ PHI(3,30),W(30),RW(30),SR(4,60),HA(3,60),HB(3,60), SLIP
     *               RPHI(3,30),HT(3,3,60),SPRING(5,90),VISC(7,90),     SETUP2
     *               JNT(30),IPIN(30),ISING(30),IGLOB(30),JOINTF(30)    SETUP2
      COMMON/CMATRX/ V1(3,30),V2(3,30),V3(3,12),B12(3,3,60),A22(3,3,60),SETUP2
     *               F(3,30),TQ(3,30),WJ(30),A11(3,3,30)                SLIP
      COMMON/CSTRNT/ A13(3,3,24),A23(3,3,24),B31(3,3,24),B32(3,3,24),   SETUP2
     *               HHT(3,3,12),RK1(3,12),RK2(3,12),QQ(3,12),TQQ(3,12),SETUP2
     *               RQQ(3,12),HQQ(3,12),SQQ(12),CFQQ(12),              SETUP2
     *               KQ1(12),KQ2(12),KQTYPE(12)                         SETUP2
      COMMON/CNSNTS/ PI,RADIAN,G,THIRD,EPS(24),                         SETUP2
     *               UNITL,UNITM,UNITT,GRAVTY(3),TWOPI                  TWOPI
      LOGICAL*1 FREE                                                    SLIP
      COMMON/TEMPVS/T(3),S(3),T1(3),T2(3),T3(3),T4(3),T5(3),T6(3),      SETUP2
     *              T7(3),T8(3),T9(3),T10(3),T11(3),T12(3),HH(3),       SETUP2
     *              TT1(3,3),TT2(3,3),S1,SQS1,S2,S3,S4                  SETUP2
     *              ,WCRM(3),RM(3),WCM(3),WWCM(3),WWM(3),RBA (3),BA     SETUP2
     *              ,WCRN(3),RN(3),WCN(3),WWCN(3),WWN(3),RBAD(3)        SETUP2
     *              ,IDUM(14290),FREE(30)                               SLIP
C                                                                       SETUP2
      CALL ELTIME(1,26)                                                 SETUP2
C                                                                       SETUP2
C     COMPUTE A22 ARRAY VIA DHHPIN FOR DAUX2 ROUTINES.                  SETUP2
C                                                                       SETUP2
      IF (NJNT.EQ.0) GO TO 50                                           SETUP2
      DO 49 M=1,NJNT                                                    SETUP2
      FREE(M) = .TRUE.                                                  SLIP
      N = IABS(JNT(M))                                                  SETUP2
      IF (N.EQ.0) GO TO 49                                              SETUP2
      IF (IPIN(M).EQ.0) GOTO 49                                         SLIP
      IF (IPIN(M).GE.2.AND. IPIN(M).LE.5) GO TO 49                      SLIP
      FREE(M) = .FALSE.                                                 SLIP
      CALL DHHPIN(A22(1,1,2*M-1),T,N  ,M,2*M-1)                         SETUP2
      CALL DHHPIN(A22(1,1,2*M  ),T,M+1,M,2*M  )                         SETUP2
   49 CONTINUE                                                          SETUP2
C                                                                       PECONV
C     THIS STATEMENT IS NECESSARY FOR THE PROGRAM TO RUN ON THE         PECONV
C         P&E FORTRAN VII O (REV 4) COMPILER                            PECONV
C                                                                       PECONV
      NNNET = IPIN(M)                                                   PECONV
C                                                                       SETUP2
```

```
C     SET UP A13,A23 AND V3 ARRAYS FOR DAUX33.                          SETUP2
C                                                                       SETUP2
   50 IF (NQ.EQ.0) GO TO 98                                             SETUP2
      DO 70 K=1,NQ                                                      SETUP2
      IF (KQTYPE(K).LT.0) GO TO 70                                      SETUP2
      IF (KQTYPE(K).EQ.5) GO TO 70                                      SETUP2
      M = KQ1(K)                                                        SETUP2
      N = KQ2(K)                                                        SETUP2
      IF (KQTYPE(K).EQ.2 .OR. KQTYPE(K).EQ.4) GO TO 53                  SETUP2
C                                                                       SETUP2
C     FOR KQTYPE = 1 OR 3, SET HHT = I                                  SETUP2
C                                                                       SETUP2
      DO 52 J=1,3                                                       SETUP2
      DO 51 I=1,3                                                       SETUP2
   51 HHT(I,J,K) = 0.0                                                  SETUP2
   52 HHT(J,J,K) = 1.0                                                  SETUP2
      IF (KQTYPE(K).NE.6) GO TO 61                                      SETUP2
C                                                                       SETUP2
C     FOR KQTYPE=6, SET HHT= I-TT'                                      SETUP2
C                                                                       SETUP2
      DO 60 J=1,3                                                       SETUP2
      DO 60 I=1,3                                                       SETUP2
   60 HHT(I,J,K) = HHT(I,J,K) - TQQ(I,K)*TQQ(J,K)                       SETUP2
      GO TO 61                                                          SETUP2
   53 IF (KQTYPE(K).NE.2) GO TO 56                                      SETUP2
C                                                                       SETUP2
C     FOR KQTYPE=2, COMPUTE HH AND HHT.                                 SETUP2
C                                                                       SETUP2
      CALL DOT31(D(1,1,M),RK1(1,K),T1)                                  SETUP2
      CALL DOT31(D(1,1,N),RK2(1,K),T2)                                  SETUP2
      S1 = 0.0                                                          SETUP2
      DO 54 I=1,3                                                       SETUP2
      HH(I) = SEGLP(I,M)+T1(I) - SEGLP(I,N)-T2(I)                       SETUP2
   54 S1 = S1 + HH(I)**2                                                SETUP2
      SQS1 = DSQRT(S1)                                                  SETUP2
      DO 55 I=1,3                                                       SETUP2
      HH(I) = HH(I)/SQS1                                                SETUP2
   55 IF (DABS(HH(I)).LE.EPS(12)) HH(I) = 0.0                           SETUP2
      CALL DOTT31(HH,HH,HHT(1,1,K))                                     SETUP2
   56 IF (KQTYPE(K).NE.4) GO TO 61                                      SETUP2
C                                                                       SETUP2
C     FOR KQTYPE = 4, SET HHT = HHT                                     SETUP2
C                                                                       SETUP2
      CALL DOTT31(HQQ(1,K),HQQ(1,K),HHT(1,1,K))                         SETUP2
C                                                                       SETUP2
C     SET A13(2K-1) =  HHT                                              SETUP2
C     AND A13(2K)   = -HHT                                              SETUP2
C                                                                       SETUP2
   61 DO 62 J=1,3                                                       SETUP2
      DO 62 I=1,3                                                       SETUP2
```

```
      A13(I,J,2*K-1) =  HHT(I,J,K)                                      SETUP2
   62 A13(I,J,2*K  ) = -HHT(I,J,K)                                      SETUP2
C                                                                       SETUP2
C     SET A23(2K-1) = (R1X)(D1)A13(2K-1)                                SETUP2
C     AND A23(2K)   = (R2X)(D2)A13(2K)                                  SETUP2
C                                                                       SETUP2
      CALL MAT33(D(1,1,M),A13(1,1,2*K-1),TT1)                           SETUP2
      CALL MAT33(D(1,1,N),A13(1,1,2*K  ),TT2)                           SETUP2
      DO 63 J=1,3                                                       SETUP2
      CALL CROSS(RK1(1,K),TT1(1,J),A23(1,J,2*K-1) )                     SETUP2
   63 CALL CROSS(RK2(1,K),TT2(1,J),A23(1,J,2*K  ) )                     SETUP2
      IF (KQTYPE(K).EQ.4) GO TO 72                                      SETUP2
C                                                                       SETUP2
C     FOR KQTYPE = 1,2 OR 3, SET B31 = A13' AND B32 = A23'              SETUP2
C                                                                       SETUP2
      DO 71 I=1,3                                                       SETUP2
      DO 71 J=1,3                                                       SETUP2
      B31(I,J,2*K-1) = A13(J,I,2*K-1)                                   SETUP2
      B31(I,J,2*K  ) = A13(J,I,2*K  )                                   SETUP2
      B32(I,J,2*K-1) = A23(J,I,2*K-1)                                   SETUP2
   71 B32(I,J,2*K  ) = A23(J,I,2*K  )                                   SETUP2
      GO TO 76                                                          SETUP2
C                                                                       SETUP2
C     FOR KQTYPE = 4, SET B31(2K-1) =  HTT                              SETUP2
C                         B31(2K  ) = -HTT                              SETUP2
C                         B32       = (B31)(D')(RX)'                    SETUP2
C                                                                       SETUP2
   72 CALL DOTT31(HQQ(1,K),TQQ(1,K),B31(1,1,2*K-1))                     SETUP2
      DO 73 I=1,3                                                       SETUP2
      DO 73 J=1,3                                                       SETUP2
   73 B31(I,J,2*K) = -B31(I,J,2*K-1)                                    SETUP2
      CALL DOTT33(D(1,1,M),B31(1,1,2*K-1),B32(1,1,2*K-1))               SETUP2
      CALL DOTT33(D(1,1,N),B31(1,1,2*K  ),B32(1,1,2*K  ))               SETUP2
      DO 74 J=1,3                                                       SETUP2
      CALL CROSS(RK1(1,K),B32(1,J,2*K-1),TT1(1,J))                      SETUP2
   74 CALL CROSS(RK2(1,K),B32(1,J,2*K  ),TT2(1,J))                      SETUP2
      DO 75 I=1,3                                                       SETUP2
      DO 75 J=1,3                                                       SETUP2
      B32(I,J,2*K-1) = TT1(J,I)                                         SETUP2
   75 B32(I,J,2*K  ) = TT2(J,I)                                         SETUP2
C                                                                       SETUP2
C     COMPUTE V3 = D2'(W2X(W2XR2)) - D1'(W1X(W1XR1))                    SETUP2
C                                                                       SETUP2
   76 CALL CROSS(WMEG(1,M),RK1(1,K),T3)                                 SETUP2
      CALL CROSS (WMEG(1,M),T3,T4)                                      SETUP2
      CALL DOT31 (D(1,1,M),T4,T5)                                       SETUP2
      CALL CROSS (WMEG(1,N),RK2(1,K),T6)                                SETUP2
      CALL CROSS (WMEG(1,N),T6,T7)                                      SETUP2
      CALL DOT31 (D(1,1,N),T7,T8)                                       SETUP2
      DO 64 I=1,3                                                       SETUP2
```

```
   64 V3(I,K) = T8(I) - T5(I)                                             SETUP2
      IF (KQTYPE(K).NE.2) GO TO 67                                        SETUP2
C                                                                         SETUP2
C      RECOMPUTE V3 FOR KQTYPE=2.                                         SETUP2
C                                                                         SETUP2
      CALL DOT31 (D(1,1,M),T3,T9 )                                        SETUP2
      CALL DOT31 (D(1,1,N),T6,T10)                                        SETUP2
      S2 = 0.0                                                            SETUP2
      DO 65 I=1,3                                                         SETUP2
      T11(I) = SEGLV(I,M)+T9(I) - SEGLV(I,N)-T10(I)                       SETUP2
   65 S2 = S2 + T11(I)**2                                                 SETUP2
      S3 = HH(1)*V3(1,K) + HH(2)*V3(2,K) + HH(3)*V3(3,K)                  SETUP2
      S4 = S3-S2/SQS1                                                     SETUP2
      DO 66 I=1,3                                                         SETUP2
   66 V3(I,K) = S4*HH(I)                                                  SETUP2
   67 IF (KQTYPE(K).NE.3.AND.KQTYPE(K).NE.6) GO TO 77                     SETUP2
C                                                                         SETUP2
C     FOR KQTYPE=3 OR 6, ADD R DOT TERM FROM PLELP OR SEGSEG TO V3.       SETUP2
C                                                                         SETUP2
      DO 68 I=1,3                                                         SETUP2
   68 V3(I,K) = V3(I,K) + RQQ(I,K)                                        SETUP2
      IF (KQTYPE(K).NE.6)  GO TO 70                                       SETUP2
C                                                                         SETUP2
C     FOR KQTYPE=6, SET V3 = (I-TT')(V3+RQQ)                              SETUP2
C                                                                         SETUP2
      VQQ = V3(1,K)*TQQ(1,K) + V3(2,K)*TQQ(2,K) + V3(3,K)*TQQ(3,K)        SETUP2
      DO 69 I=1,3                                                         SETUP2
   69 V3(I,K) = V3(I,K) - VQQ*TQQ(I,K)                                    SETUP2
   77 IF (KQTYPE(K).NE.4) GO TO 70                                        SETUP2
C                                                                         SETUP2
C     FOR KQTYPE = 4, ADD R TERM FROM PLELP OR SEGSEG TO V3.              SETUP2
C                                                                         SETUP2
      S3 = TQQ(1,K)*V3(1,K) + TQQ(2,K)*V3(2,K) + TQQ(3,K)*V3(3,K)         SETUP2
      S4 = S3+SQQ(K)                                                      SETUP2
      DO 78 I=1,3                                                         SETUP2
   78 V3(I,K) = S4*HQQ(I,K)                                               SETUP2
   70 CONTINUE                                                            SETUP2
C                                                                         SETUP2
C     SPECIAL SETUP FOR TENSION ELEMENTS (KQTYPE = 5).                    SETUP2
C                                                                         SETUP2
      N = 0                                                               SETUP2
   79 N = N+1                                                             SETUP2
      IF (N.GE.NQ) GO TO 98                                               SETUP2
      IF (KQTYPE(N).NE.5) GO TO 79                                        SETUP2
      DO 81 I=1,3                                                         SETUP2
      DO 80 J=1,3                                                         SETUP2
      A13(I,J,2*N-1) = 0.0                                                SETUP2
      A13(I,J,2*N  ) = 0.0                                                SETUP2
      A23(I,J,2*N  ) = 0.0                                                SETUP2
      B31(I,J,2*N-1) = 0.0                                                SETUP2
```

```
      B31(I,J,2*N  ) = 0.0                                              SETUP2
      A13(I,J,2*N+1) = 0.0                                              SETUP2
      A13(I,J,2*N+2) = 0.0                                              SETUP2
      A23(I,J,2*N+1) = 0.0                                              SETUP2
      B31(I,J,2*N+1) = 0.0                                              SETUP2
      B31(I,J,2*N+2) = 0.0                                              SETUP2
      HHT(I,J,N    ) = 0.0                                              SETUP2
   80 HHT(I,J,N+1  ) = 0.0                                              SETUP2
      A13(I,I,2*N-1) = 1.0                                              SETUP2
      B31(I,I,2*N-1) = RK1(1,N+1)                                       SETUP2
      B31(I,I,2*N  ) = RK1(3,N+1)                                       SETUP2
      A13(I,I,2*N+2) = 1.0                                              SETUP2
      B31(I,I,2*N+1) = RK1(3,N+1)                                       SETUP2
   81 B31(I,I,2*N+2) = RK1(2,N+1)                                       SETUP2
      N1 = KQ1(N)                                                       SETUP2
      N2 = KQ2(N)                                                       SETUP2
      DO 82 K=1,3                                                       SETUP2
      CALL CROSS(RK1(1,N),D(1,K,N1),A23(1,K,2*N-1))                     SETUP2
   82 CALL CROSS(RK2(1,N),D(1,K,N2),A23(1,K,2*N+2))                     SETUP2
      DO 83 I=1,3                                                       SETUP2
      DO 83 J=1,3                                                       SETUP2
      B32(I,J,2*N-1) = RK1(1,N+1)*A23(J,I,2*N-1)                        SETUP2
      B32(I,J,2*N  ) = RK1(3,N+1)*A23(J,I,2*N+2)                        SETUP2
      B32(I,J,2*N+1) = RK1(3,N+1)*A23(J,I,2*N-1)                        SETUP2
   83 B32(I,J,2*N+2) = RK1(2,N+1)*A23(J,I,2*N+2)                        SETUP2
      CALL CROSS(WMEG(1,N1),RK1(1,N),WCRM)                              SETUP2
      CALL CROSS(WMEG(1,N2),RK2(1,N),WCRN)                              SETUP2
      CALL DOT31(D(1,1,N1),RK1(1,N),RM)                                 SETUP2
      CALL DOT31(D(1,1,N2),RK2(1,N),RN)                                 SETUP2
      CALL DOT31(D(1,1,N1),WCRM,WCM)                                    SETUP2
      CALL DOT31(D(1,1,N2),WCRN,WCN)                                    SETUP2
      BA = 0.0                                                          SETUP2
      DO 84 I=1,3                                                       SETUP2
      RBA (I) = SEGLP(I,N2) + RN (I) - SEGLP(I,N1) - RM (I)             SETUP2
      RBAD(I) = SEGLV(I,N2) + WCN(I) - SEGLV(I,N1) - WCM(I)             SETUP2
   84 BA = BA + RBA(I)**2                                               SETUP2
      BA = DSQRT(BA)                                                    SETUP2
      FORCE = 0.0                                                       SETUP2
      IF (BA.GT.RK2(3,N+1)) FORCE = RK2(1,N+1)*(1.0-RK2(3,N+1)/BA)      SETUP2
      DO 85 I=1,3                                                       SETUP2
      V3(I,N) = RK2(2,N+1)*RBAD(I) + FORCE*RBA(I)                       SETUP2
   85 V3(I,N+1) = -V3(I,N)                                              SETUP2
      CALL CROSS(WMEG(1,N1),WCRM,WWCM)                                  SETUP2
      CALL CROSS(WMEG(1,N2),WCRN,WWCN)                                  SETUP2
      CALL DOT31(D(1,1,N1),WWCM,WWM)                                    SETUP2
      CALL DOT31(D(1,1,N2),WWCN,WWN)                                    SETUP2
      DO 86 I=1,3                                                       SETUP2
      V3(I,N  ) = V3(I,N  ) - RK1(1,N+1)*WWM(I) - RK1(3,N+1)*WWN(I)     SETUP2
   86 V3(I,N+1) = V3(I,N+1) - RK1(3,N+1)*WWM(I) - RK1(2,N+1)*WWN(I)     SETUP2
      N = N+1                                                           SETUP2
```

```
      GO TO 79                                                          SETUP2
   98 CALL ELTIME(2,26)                                                 SETUP2
      RETURN                                                            SETUP2
      END                                                               SETUP2
```

```
      SUBROUTINE SINPUT                                                  SINPUT
C                                                     REV IV    02/20/87HYPER
C     READS AND PRINTS THE INPUT CARDS THAT DESCRIBE THE PHYSICAL        SINPUT
C     DIMENSIONS OF THE PLANES REPRESENTING THE VEHICLE PANELS AND OF    SINPUT
C     THE RESTRAINT BELTS. ALSO PROCESSES THOSE DATA CARDS THAT DESCRIBESINPUT
C     ADDITIONAL CONTACT ELLIPSOIDS, CONSTRAINTS, BODY SEGMENT SYMMETRY SINPUT
C     OPTIONS AND SPRING DAMPER FUNCTIONS.                               SINPUT
C                                                                        SINPUT
      IMPLICIT REAL*8 (A-H,O-Z)                                          SINPUT
      COMMON/CONTRL/ TIME,NSEG,NJNT,NPL,NBLT,NBAG,NVEH,NGRND,            SINPUT
     *               NS,NQ,NSD,NFLX,NHRNSS,NWINDF,NJNTF,NPRT(36),NPG     PAGE
      COMMON/CNTSRF/ PL(24,30),BELT(20,8),TPTS(6,8),BD(24,40)            EDGE
      COMMON/SGMNTS/ D(3,3,30),WMEG(3,30),WMEGD(3,30),U1(3,30),U2(3,30),SINPUT
     *               SEGLP(3,30),SEGLV(3,30),SEGLA(3,30),NSYM(30)        SINPUT
      COMMON/CSTRNT/ A13(3,3,24),A23(3,3,24),B31(3,3,24),B32(3,3,24),    SINPUT
     *               HHT(3,3,12),RK1(3,12),RK2(3,12),QQ(3,12),TQQ(3,12),SINPUT
     *               RQQ(3,12),HQQ(3,12),SQQ(12),CFQQ(12),               SINPUT
     *               KQ1(12),KQ2(12),KQTYPE(12)                          SINPUT
      COMMON/TITLES/ DATE(3),COMENT(40),VPSTTL(20),BDYTTL(5),            SINPUT
     *               BLTTTL(5,8),PLTTL(5,30),BAGTTL(5,6),SEG(30),        SINPUT
     *               JOINT(30),CGS(30),JS(30)                            SINPUT
      REAL DATE,COMENT,VPSTTL,BDYTTL,BLTTTL,PLTTL,BAGTTL,SEG,JOINT       SINPUT
      LOGICAL*1 CGS,JS,LP4                                               HYPER
      COMMON/DAMPER/ APSDM(3,20),APSDN(3,20),ASD(5,20),MSDM(20),MSDN(20)SINPUT
      COMMON/WINDFR/ WTIME(30),QFU(3,5),QFV(3,5),WF(3,30),IWIND(30),     WINDOP
     *               MWSEG(7,30),NFVSEG(6),NFVNT(5),MOWSEG(30,30)        WINDOP
      COMMON/CNSNTS/ PI,RADIAN,G,THIRD,EPS(24),                          SINPUT
     *               UNITL,UNITM,UNITT,GRAVTY(3),TWOPI                   TWOPI
      COMMON/TEMPVS/ P1(3),P2(3),P3(3),P4(3),DE(3,3)                     HYPER
      DIMENSION IDYPR(3)                                                 SINPUT
      DATA IDYPR/3,2,1/                                                  SINPUT
      DATA MAXBD/40/                                                     CHGIII
      DATA NPLMAX/30/,NBLTMX/8/,NBAGMX/5/,NELPMX/40/,NQMAX/12/,          MISC
     *     NSDMAX/20/,NHRNSM/5/,NWINDM/50/,NJNTFM/50/,NFORCM/5/          MISC
C                                                                        SINPUT
C     INPUT CARD D.1                                                     SINPUT
C                                                                        SINPUT
      READ  (5,11)  NPL,NBLT,NBAG,NELP,NQ,NSD,NHRNSS,NWINDF,NJNTF,NFORCESINPUT
   11 FORMAT(12I6)                                                       SINPUT
      WRITE (6,16) NPG,NPL,NBLT,NBAG,NELP,NQ,NSD,NHRNSS,NWINDF,NJNTF,    PAGE
     *             NFORCE                                                PAGE
      NPG=NPG+1                                                          PAGE
   16 FORMAT('1     NPL     NBLT     NBAG     NELP       NQ      NSD  NHRNSS',PAGE
     *      '  NWINDF   NJNTF   NFORCE',43X,'PAGE',I5/10I8,40X,'CARD D.1')PAGE
      IF (NPL.GT.NPLMAX) STOP 65                                         CHGIII
      IF (NBLT.GT.NBLTMX) STOP 66                                        MISC
      IF (NBAG.GT.NBAGMX) STOP 67                                        MISC
      IF (NELP.GT.NELPMX) STOP 68                                        MISC
      IF (NQ.GT.NQMAX) STOP 69                                           CHGIII
      IF (NSD.GT.NSDMAX) STOP 70                                         CHGIII
```

```
      IF (NHRNSS.GT.NHRNSM) STOP 71                                     MISC
      IF (NWINDF.GT.NWINDM) STOP 72                                     MISC
      IF (NJNTF.GT.NJNTFM) STOP 73                                      MISC
      IF (NFORCE.GT.NFORCM) STOP 74                                     MISC
      IF (NPL.EQ.0) GO TO 15                                            SINPUT
      IPAGE = 0                                                         SINPUT
      DO 20 J=1,NPL                                                     SINPUT
C                                                                       SINPUT
C     READ AND PRINT CARDS D.2.A,D.2.B AND D.2.C FOR THE JTH PLANE.     SINPUT
C                                                                       SINPUT
      READ (5,23) JJ,(PLTTL(I,J),I = 1,5),P1,P2,P3                      SINPUT
   23 FORMAT (I4,4X,5A4/(3F12.0))                                       SINPUT
      IF (JJ.NE.J) WRITE (6,24) JJ,J                                    SINPUT
   24 FORMAT (' PLANE INDEX INPUT ERROR,',2I4)                          SINPUT
      IF  (JJ.NE.J)  STOP 10                                            SINPUT
      IF (MOD(J,7).EQ.1.AND.IPAGE.EQ.0) WRITE (6,12) IPAGE              PAGE
      IF (MOD(J,7).EQ.1.AND.IPAGE.EQ.1) WRITE (6,112) IPAGE,NPG         PAGE
      IF (MOD(J,7).EQ.1.AND.IPAGE.EQ.1) NPG=NPG+1                       PAGE
  112 FORMAT(I1,' PLANE INPUTS',109X,'PAGE',I5/120X,'CARDS D.2')        PAGE
   12 FORMAT(I1,' PLANE INPUTS',106X,'CARDS D.2')                       SINPUT
      IPAGE = 1                                                         SINPUT
      WRITE (6,25) J, (PLTTL(I,J),I = 1,5),P1,P2,P3                     SINPUT
   25 FORMAT('0 PLANE NO.',I4,4X,5A4//17X,'X',11X,'Y',11X,'Z'/          SINPUT
     *         '  POINT 1 ' ,3F12.4/                                    SINPUT
     *         '  POINT 2 ' ,3F12.4/                                    SINPUT
     *         '  POINT 3 ' ,3F12.4)                                    SINPUT
C                                                                       SINPUT
C     PROGRAM NOW ASSUMES THE FINITE PLANE IS A PARALLELOGRAM IN SHAPE  SINPUT
C     WHERE THE INPUT POINTS P1,P2,P3 ARE 3 OF THE CORNERS SUCH THAT    SINPUT
C     EDGE P1-P2 IS LESS THAN 180 DEGREES CLOCKWISE (AS VIEWED BY THE   SINPUT
C     OCCUPANT) FROM THE EDGE P1-P3.                                    SINPUT
C                                                                       SINPUT
C     SET UP PL ARRAY AS REQUIRED BY SUBROUTINE PLELP                   SINPUT
C                                                                       SINPUT
C         PL(1,J) = A0    NORMAL EQUATION OF JTH PLACE                  SINPUT
C         PL(2,J) = B0      A0*X + B0*Y + C0*Z = D0                     SINPUT
C         PL(3,J) = C0                                                  SINPUT
C         PL(4,J) = D0                                                  SINPUT
C                                                                       SINPUT
C         PL(5,J)                                                       SINPUT
C         PL(6,J)         POINT 1                                       EDGE
C         PL(7,J)                                                       SINPUT
C                                                                       SINPUT
C         PL(8,J) =A1                                                   SINPUT
C         PL(9,J) =B1     NORMAL EQUATION OF 1ST BOUNDARY PLANE         SINPUT
C         PL(10,J)=C1       A1*X + B1*Y + C1*Z = D1                     SINPUT
C         PL(11,J)=D1     AND E1 IS LENGTH OF PLANE FROM BOUNDARY.      SINPUT
C         PL(12,J)=E1                                                   SINPUT
C                                                                       SINPUT
C         PL(13,J)=A2                                                   SINPUT
```

```
C         PL(14,J)=B2    NORMAL EQUATION OF 2ND BOUNDARY PLANE           SINPUT
C         PL(15,J)=C2      A2*X + B2*Y + C2*Z = D2                       SINPUT
C         PL(16,J)=D2    AND E2 IS LENGTH OF PLANE FROM BOUNDARY.        SINPUT
C         PL(17,J)=E2                                                    SINPUT
C                                                                        SINPUT
C         PL(18,J)                                                       EDGE
C         PL(19,J)       POINT 2 - POINT 1                               EDGE
C         PL(20,J)                                                       EDGE
C                                                                        EDGE
C         PL(21,J)                                                       EDGE
C         PL(22,J)       POINT 3 - POINT 1                               EDGE
C         PL(23,J)                                                       EDGE
C                                                                        EDGE
C         PL(24,J)       NOT CURRENTLY USED                              EDGE
      S22 = 0.0                                                          SINPUT
      S23 = 0.0                                                          SINPUT
      S33 = 0.0                                                          SINPUT
      DO 26 I =1,3                                                       SINPUT
      P2(I) = P2(I)-P1(I)                                                SINPUT
      P3(I) = P3(I)-P1(I)                                                SINPUT
      PL(I+ 4,J) = P1(I)                                                 EDGE
      PL(I+17,J) = P2(I)                                                 EDGE
      PL(I+20,J) = P3(I)                                                 EDGE
      S22 = S22 + P2(I)*P2(I)                                            SINPUT
      S23 = S23 + P2(I)*P3(I)                                            SINPUT
   26 S33 = S33 + P3(I)*P3(I)                                            SINPUT
      S2  = DSQRT(S22)                                                   SINPUT
      S3  = DSQRT(S33)                                                   SINPUT
      CALL CROSS(P2,P3,PL(1,J))                                          SINPUT
      S1 = 0.0                                                           SINPUT
      DO 27 I=1,3                                                        SINPUT
   27 S1 = S1 + PL(I,J)**2                                               SINPUT
      S1 = DSQRT(S1)                                                     SINPUT
      DO 28 I=1,3                                                        SINPUT
      PL(I,J) = PL(I,J)/S1                                               SINPUT
      PL(I+7 ,J) = (S33*P2(I) - S23*P3(I)) / (S1*S3)                     SINPUT
   28 PL(I+12,J) = (S22*P3(I) - S23*P2(I)) / (S1*S2)                     SINPUT
      PL( 4,J) = P1(1)*PL( 1,J) + P1(2)*PL( 2,J) + P1(3)*PL( 3,J)        SINPUT
      PL(11,J) = P1(1)*PL( 8,J) + P1(2)*PL( 9,J) + P1(3)*PL(10,J)        SINPUT
      PL(12,J) = P2(1)*PL( 8,J) + P2(2)*PL( 9,J) + P2(3)*PL(10,J)        SINPUT
      PL(16,J) = P1(1)*PL(13,J) + P1(2)*PL(14,J) + P1(3)*PL(15,J)        SINPUT
   20 PL(17,J) = P3(1)*PL(13,J) + P3(2)*PL(14,J) + P3(3)*PL(15,J)        SINPUT
   15 IF (NBLT.EQ.0) GO TO 35                                            SINPUT
      DO 30 J=1,NBLT                                                     SINPUT
C                                                                        SINPUT
C     READ AND PRINT CARDS D.3.A, D.3.B AND D.3.C FOR THE JTH BELT.      SINPUT
C                                                                        SINPUT
      READ (5,13) (BLTTTL(I,J),I = 1,5),(BELT(I,J),I = 1,11)             SINPUT
   13 FORMAT (5A4/(6F12.0))                                              SINPUT
      IF (MOD(J,5).EQ.1) WRITE (6,21) NPG                                PAGE
```

```
      IF (MOD(J,5).EQ.1) NPG=NPG+1                                      PAGE
   21 FORMAT('1 BELT INPUTS',110X,'PAGE',I5/120X,'CARDS D.3')           PAGE
   30 WRITE (6,14) J,(BLTTTL(I,J),I = 1,5),(BELT(I,J),I = 1,11)         SINPUT
   14 FORMAT('0 BELT NO.',I4,4X,5A4//                                   SINPUT
     *    30X,'ANCHOR POINT A',46X,'ANCHOR POINT B'/                    SINPUT
     *    2(16X,'X',19X,'Y',19X,'Z',3X)/6F20.3//                        SINPUT
     *    26X,'FIXED POINT ON SEGMENT',45X,'SLACK(+)'/                  SINPUT
     *    16X,'X',19X,'Y',19X,'Z',17X,'BLANK',13X,'LENGTH(-)'/5F20.3)   SINPUT
C                                                                       SINPUT
C     CALL AIRBG1 ROUTINE IF REQUIRED FOR AIRBAG INPUT                  SINPUT
C                                                                       SINPUT
   35 IF (NBAG.NE.0) CALL AIRBG1                                        SINPUT
      IF (NELP.LE.0) GO TO 51                                           SINPUT
C                                                                       SINPUT
C     READ AND PRINT CARDS D.5 FOR ELLIPSOID INPUT, IF ANY.             SINPUT
C       NOTE: NELP IS THE NO. OF ELLIPSOIDS TO BE SUPPLIED HERE, NOT THESINPUT
C             NO. OF ELLIPSOIDS IN THE PROGRAM, SINCE THE FIRST NSEG    SINPUT
C             ELLIPSOIDS WERE SUPPLIED ON CARDS B.2.A - B.2.I. HOWEVER  SINPUT
C             THEY MAY BE REPLACED HERE IF DESIRED.                     SINPUT
C                                                                       SINPUT
      WRITE (6,41) NPG,UNITL,UNITL                                      PAGE
      NPG=NPG+1                                                         PAGE
   41 FORMAT('1 ADDITIONAL ELLIPSOID INPUT',95X,'PAGE',I5/120X,         PAGE
     *  'CARDS D.5'/17X,'SEMIAXES (',A4,')',18X,'OFFSET (',A4,')',      PAGE
     *  20X,'ROTATION (DEG)',15X,'POWER'/                               HYPER
     * .3X,'NO.',2(8X,'X',8X,'Y',8X,'Z',6X),7X,'YAW',7X,'PITCH',5X,     SINPUT
     *  'ROLL'//)                                                       SINPUT
      DO 50 MM=1,NELP                                                   SINPUT
      READ (5,42) M,P1,P2,P3,P4                                         HYPER
   42 FORMAT(I6,9F6.0,3F4.0)                                            HYPER
      IF (M.GT.MAXBD) STOP 63                                           CHGIII
C                                                                       CHGIII
C     PREVENT EXTRA ELLIPSOIDS FROM CHANGING AIRBAG ELLIPSOIDS          CHGIII
C                                                                       CHGIII
      IF (M.GT.NVEH.AND.M.LT.NGRND) WRITE (6,330)                       CHGIII
  330 FORMAT(3X,'THE EXTRA CONTACT ELLIPSOID NUMBER IS THE SAME AS AN AICHGIII
     *RBAG ELLIPSOID')                                                  CHGIII
      IF (M.GT.NVEH.AND.M.LT.NGRND) STOP 64                             CHGIII
      WRITE (6,43) M,P1,P2,P3,P4                                        HYPER
   43 FORMAT(I6,3(3X,3F9.3,3X),3F6.0)                                   HYPER
      CALL DRCYPR (DE,P3,IDYPR)                                         SINPUT
      N = 1                                                             HYPER
      LP4 = .FALSE.                                                     HYPER
      DO 39 J = 1,3                                                     HYPER
   39 IF (P4(J).GT.2.0) LP4 = .TRUE.                                    HYPER
      IF (LP4) N = 2                                                    HYPER
      DO 46 I = 1,3                                                     HYPER
      BD(N  ,M) = P1(I)                                                 HYPER
      BD(N+3,M) = P2(I)                                                 HYPER
      IF (LP4) GO TO 46                                                 HYPER
```

```
      DO 45 J=1,3                                                       SINPUT
      SUM1 = 0.0                                                        SINPUT
      SUM2 = 0.0                                                        SINPUT
      DO 44 L=1,3                                                       SINPUT
      SUM1 = SUM1 + DE(L,I)/P1(L)**2*DE(L,J)                            SINPUT
   44 SUM2 = SUM2 + DE(L,I)*P1(L)**2*DE(L,J)                            SINPUT
      K = 3*I +J +3                                                     SINPUT
      BD(K   ,M) = SUM1                                                 SINPUT
   45 BD(K+9,M) = SUM2                                                  SINPUT
   46 N = N + 1                                                         HYPER
      IF (.NOT.LP4) GO TO 50                                            HYPER
      BD(1,M) = -P4(1)                                                  HYPER
      N = 8                                                             HYPER
      DO 48 J = 1,3                                                     HYPER
      BD(J+19,M) = P4(J)                                                HYPER
      IF (BD(J+19,M).EQ.0.0) BD(J+19,M) = BD(20,M)                      HYPER
      BD(J+16,M) = 1.0/BD(J+1,M)**2                                     HYPER
      DO 48 I = 1,3                                                     HYPER
      BD(N,M) = DE(I,J)                                                 HYPER
   48 N = N + 1                                                         HYPER
      BD(23,M) = 0.0                                                    HYPER
      IF (BD(20,M).NE.BD(21,M)) BD(23,M) = 1.0                          HYPER
      IF (BD(21,M).NE.BD(22,M)) BD(23,M) = 1.0                          HYPER
      IF (BD(22,M).NE.BD(20,M)) BD(23,M) = 1.0                          HYPER
   50 CONTINUE                                                          SINPUT
C                                                                       SINPUT
C     READ AND PRINT CARDS D.6 FOR CONSTRAINT INPUT, IF ANY.            SINPUT
C                                                                       SINPUT
   51 IF (NQ.LE.0) GO TO 70                                             SINPUT
      DO 60 K=1,NQ                                                      SINPUT
      READ (5,52) KQTYPE(K),KQ1(K),KQ2(K),(RK1(I,K),I=1,3)              SINPUT
     *                                   ,(RK2(I,K),I=1,3)              SINPUT
   52 FORMAT(3I6,6F6.0)                                                 SINPUT
      IF (K.EQ.1) WRITE (6,53) NPG,UNITL,UNITL                          PAGE
      IF (K.EQ.1) NPG=NPG+1                                             PAGE
   53 FORMAT('1 CONSTRAINT INPUT',105X,'PAGE',I5/120X,'CARDS D.6'/      PAGE
     *       '  TYPE   SEGMENT  SEGMENT       POINT ON 1ST SEGMENT (',  SINPUT
     *       A4,')','      POINT ON 2ND SEGMENT (',A4,')'/              SINPUT
     *       '    NO.     NO. 1    NO. 2          X          Y         Z    SINPUT
     *          X          Y          Z'//)                             SINPUT
      WRITE (6,54)  KQTYPE(K),KQ1(K),KQ2(K),(RK1(I,K),I=1,3)            SINPUT
     *                                   ,(RK2(I,K),I=1,3)              SINPUT
   54 FORMAT(I6,2I9,2(6X,3F9.3) )                                       SINPUT
   60 CONTINUE                                                          SINPUT
C                                                                       SINPUT
C     CARD D.7  BODY SEGMENT SYMMETRY INPUT                             SINPUT
C                                                                       SINPUT
   70 READ (5,71) (NSYM(J),J=1,NSEG)                                    SINPUT
   71 FORMAT(18I4)                                                      SINPUT
      DO 103 J=1,NSEG                                                   TGMOD2
```

```
      LJ = NSYM(J)                                                     TGMOD2
      IF(IABS(LJ).GT.NSEG) GO TO 107                                   TGMOD2
      IF(LJ) 104,103,105                                               TGMOD2
  105 LK = NSYM(LJ)                                                    TGMOD2
      IF(IABS(LK).GT.NSEG) GO TO 107                                   TGMOD2
      IF(LK.NE.J) GO TO 106                                            TGMOD2
      GO TO 103                                                        TGMOD2
  104 JJ = -J                                                          TGMOD2
      LJ = -LJ                                                         TGMOD2
      LK = NSYM(LJ)                                                    TGMOD2
      IF(IABS(LK).GT.NSEG) GO TO 107                                   TGMOD2
      IF((LK.NE.JJ).OR.(NSYM(J).EQ.JJ)) GO TO 106                      TGMOD2
      GO TO 103                                                        TGMOD2
  106 STOP 96                                                          TGMOD2
  107 STOP 97                                                          TGMOD2
  103 CONTINUE                                                         TGMOD2
      WRITE(6,72) (J,J=1,NSEG)                                         SINPUT
      WRITE(6,73) (NSYM(J),J=1,NSEG)                                   SINPUT
   72 FORMAT('0 BODY SEGMENT SYMMETRY INPUT',91X,'CARD D.7'//          SINPUT
     *       ' SEG NO.',30I4)                                          SINPUT
   73 FORMAT('0 NSYM(J)',30I4)                                         SINPUT
      NSEG1 = NSEG+1                                                   SINPUT
      DO 74 J=NSEG1,NGRND                                              SINPUT
   74 NSYM(J) = 0                                                      SINPUT
      IF  (NSD.LE.0)  GO TO 90                                         SINPUT
C                                                                      SINPUT
C     CARD D.8   SPRING DAMPERS FUNCTION INPUT.                        SINPUT
C                                                                      SINPUT
      DO 79 J=1,NSD                                                    SINPUT
   79 READ (5,80) MSDM(J),MSDN(J),(APSDM(I,J),I=1,3),                  SINPUT
     *            (APSDN(I,J),I=1,3),(ASD(I,J),I=1,5)                  SINPUT
   80 FORMAT(2I3,11F6.0)                                               SINPUT
      WRITE (6,81) UNITL                                               SINPUT
   81 FORMAT('0',5X,'SPRING  DAMPERS  FUNCTION  INPUT',82X,'CARDS D.8'//SINPUT
     *   18X,'COORDINATES OF ATTACHMENT POINTS (',A4,')'/              SINPUT
     *   5X,'SEGMENT',9X,'SEGMENT M',16X,'SEGMENT N',15X,              SINPUT
     *   'SPRING FORCE FUNCTION',12X,'DAMPING FORCE FUNCTION'/         AFREVS
     *   ' NO.  M    N',2(6X,'X',7X,'Y',7X,'Z',2X),7X,'D0',9X,'A1',11X, SINPUT
     *   'A2',13X,'B1',10X,'B2' // )                                   SINPUT
      DO  82  J=1,NSD                                                  SINPUT
   82 WRITE (6,83) J,MSDM(J),MSDN(J),(APSDM(I,J),I=1,3),               SINPUT
     *              (APSDN(I,J),I=1,3),(ASD(I,J),I=1,5)                SINPUT
   83 FORMAT(I3,2I4,2(1X,3F8.2),F11.2,2F12.3,F15.3,F12.3)              SINPUT
C                                                                      SINPUT
C     CARDS D.9  FORCE AND/OR TORQUE FUNCTIONS.                        CHGIII
C                                                                      SINPUT
   90 NFVSEG(6)= NFORCE                                                SINPUT
      IF  (NFORCE.LE.0)  GO TO 99                                      SINPUT
      WRITE  (6,91)                                                    SINPUT
   91 FORMAT ('0',6X,'FORCE AND/OR TORQUE FUNCTION INPUTS',78X,'CARDS D.CHGIII
```

```
     *9'//,    5X,'NO.', 5X,'SEG', 5X,'FCN', 13X,'X', 9X,'Y', 9X,'Z',      CHGIII
     *         13X,'YAW', 6X,'PITCH', 6X,'ROLL' //)                         SINPUT
      DO  95  J=1,NFORCE                                                    SINPUT
      READ  (5,92)  NFVSEG(J),NFVNT(J),P1,P2                                SINPUT
   92 FORMAT (2I6,6F10.0)                                                   SINPUT
      WRITE  (6,93)  J,NFVSEG(J),NFVNT(J),P1,P2                             SINPUT
   93 FORMAT (3I8,6X,3F10.3,6X,3F10.3)                                      SINPUT
      CALL DRCYPR (DE,P2,IDYPR)                                             SINPUT
      DO  94  I=1,3                                                         SINPUT
   94 QFU(I,J) = DE(1,I)                                                    FIXSPT
   95 CALL CROSS (P1,QFU(1,J),QFV(1,J))                                     SINPUT
   99 RETURN                                                                SINPUT
      END                                                                   SINPUT
```

```
      SUBROUTINE  SLPLOT  (X, NX, X0, XN, XL, XSIZE, XLAB, NXLB,          SLPLOT
     *                     Y, NY, Y0, YN, YL, YSIZE, YLAB, NYLB,          SLPLOT
     *                     NPTS, NYY, NDY, PLAB1, NPLB1, PLAB2, NPLB2)    SLPLOT
C                                                 REV III.2 08/08/84REVIII
C                                                                         SLPLOT
C     ARGUMENTS:                                                          SLPLOT
C        X(NPTS)      - ARRAY OF NPTS ABSCISSAS TO BE PLOTTED.            SLPLOT
C        Y(NDY,NYY)   - ARRAY OF NPTS*NYY ORDINATES TO BE PLOTTED.        SLPLOT
C        NX,NY        - POSITIVE - NO. OF LINEAR SUBDIVISIONS.            SLPLOT
C                       NEGATIVE - NO. OF LOGARITHMIC DECADES.            SLPLOT
C        X0,Y0        - AXES ORIGINS (POWER OF TEN IF NX,NY NEGATIVE).    SLPLOT
C        XN,YN        - AXES END VALUES (REQUIRED IF NX,NY POSITIVE).     SLPLOT
C        XL,YL        - LENGTH (INCHES) OF X,Y AXES.                      SLPLOT
C        XSIZE,YSIZE - PAPER SIZE (INCHES) IN X,Y DIRECTIONS.             SLPLOT
C        XLAB,YLAB    - X,Y AXES LABELS (ALPHANUMERIC ARRAYS).            SLPLOT
C        NXLB,NYLB    - NO. OF CHARACTERS IN X,Y LABELS.                  SLPLOT
C        NPTS         - NO. OF POINTS IN X ARRAY AND EACH Y ARRAY.        SLPLOT
C        NYY          - NO. OF Y ARRAYS TO BE PLOTTED VS. X ARRAY.        SLPLOT
C        NDY          - FIRST DIMENSION OF Y ARRAY IN CALLING ROUTINE.    SLPLOT
C                       (NDY MUST BE .GE. NPTS)                           SLPLOT
C        PLAB1,PLAB2 - 1ST & 2ND LINES OF PLOT ID LABELS (ALPHANUMERIC).SLPLOT
C        NPLB1,NPLB2 - NO. OF CHARACTERS IN PLOT ID LABELS.               SLPLOT
C                                                                         SLPLOT
C     NOTE: PLOTS WILL BE TRUNCATED AS FOLLOWS:                           SLPLOT
C      NX,NY POSITIVE - X0,Y0 .LE. X,Y .LE. XN,YN                         SLPLOT
C      NX,NY NEGATIVE - X0,Y0 .LE. X,Y .LE. XN*10**(-NX),Y0*10**(-NY)     SLPLOT
C                                                                         SLPLOT
      DIMENSION X(NPTS),Y(NDY,NYY),XLAB(1),YLAB(1),PLAB1(1),PLAB2(1)      SLPLOT
C                                                                         SLPLOT
C     NOTE: THIS ROUTINE HAS BEEN WRITTEN FOR THE PLOTTING FACILITIES     SLPLOT
C     AT CALSPAN. THE FOLLOWING ITEMS ARE KNOWN TO BE CONTRARY TO THE     SLPLOT
C     NORMAL CALCOMP PROCEDURES AND SHOULD BE EXAMINED BY USERS AT OTHERSLPLOT
C     COMPUTER SYSTEMS AND CHANGES MADE ACCORDINGLY.                      SLPLOT
C                                                                         SLPLOT
C       1. AT CALSPAN THE PLOTTED CHARACTERS GENERATED BY SUBROUTINE      SLPLOT
C          SYMBOL HAVE A WIDTH OF 6/7 TIMES THE HEIGHT. FOR THE CALCOMP SLPLOT
C          ROUTINES THE WIDTH IS EQUAL TO THE HEIGHT. THE STATEMENT       SLPLOT
C          'WIDTHF = 6.0/7.0' SHOULD BE CHANGED TO 'WIDTHF = 1.0'.        SLPLOT
C                                                                         SLPLOT
C       2. THE ONLY INITIALIZATION REQUIRED AT CALSPAN IS THE STATEMENT SLPLOT
C          'CALL PLOT (0.0,0.0,0)' TO ESTABLISH A NEW PAGE, INCLUDING     SLPLOT
C          THE FIRST PAGE. THIS IS FOLLOWED BY 'CALL PLOT (X0,Y0,-3)' TOSLPLOT
C          SET THE PLOT ORIGIN ON THE PAGE. PROPER PLOT INITIALIZATION    SLPLOT
C          SHOULD BE DONE HERE AND IN SUBROUTINE POSTPR (AFTER STATEMENTSLPLOT
C          NO. 30) AS REQUIRED BY THE USER'S PLOTTING FACILITY.           SLPLOT
C                                                                         SLPLOT
C       3. THE STATEMENT 'CALL NEWPEN(2)' SHOULD BE EXAMINED OR DELETED.SLPLOT
C                                                                         SLPLOT
C       4. THE STATEMENT 'CALL EFPLOT' AFTER STATEMENT NO. 50 IN POSTPR SLPLOT
C          IS REQUIRED AT CALSPAN TO CLOSE OUT THE PLOT FILES. THIS       SLPLOT
```

```
C          SHOULD BE CHANGED TO CONFORM TO THE REQUIREMENTS OF THE      SLPLOT
C          USER'S PLOTTING FACILITIES.                                   SLPLOT
C                                                                        SLPLOT
C       5. THE NECESSARY JOB CONTROL LANGUAGE FOR PLOTTING IS NECESSARY.SLPLOT
C                                                                        SLPLOT
C       6. THE ONLY CALCOMP ROUTINES NEEDED ARE SYMBOL, NUMBER AND PLOT.SLPLOT
C                                                                        SLPLOT
      LOGICAL NXPOS,NXNEG,NYPOS,NYNEG                                    SLPLOT
      DATA HN/0.07/, HL/0.105/                                           SLPLOT
      WIDTHF = 1.0                                                       REDIMN
      WN = WIDTHF*HN                                                     SLPLOT
      WL = WIDTHF*HL                                                     SLPLOT
C         **  PLOT PAGE INITIALIZATION  **                               SLPLOT
      CALL PLOT (0.0,0.0,-3)                                             CHANGE
      XP =  0.5*(XSIZE-(XL-0.5))                                         SLPLOT
      YP = 0.5*(YSIZE-(YL-1.0))                                          SLPLOT
      CALL PLOT (XP,YP,-3)                                               SLPLOT
      NXPOS = NX.GT.0                                                    SLPLOT
      NXNEG = NX.LT.0                                                    SLPLOT
      NYPOS = NY.GT.0                                                    SLPLOT
      NYNEG = NY.LT.0                                                    SLPLOT
C         **  PLOT AXES AND ID LABELS.  **                               SLPLOT
      XP = 0.0                                                           SLPLOT
      YP = 0.0                                                           SLPLOT
      IF  ( NOT.NXPOS)  GO TO 12                                         SLPLOT
C         **        LINEAR X AXIS       **                               SLPLOT
      CALL LINAXS (XP, YP, 0.0, NX, XL)                                  SLPLOT
      XB = XL/(XN-X0)                                                    SLPLOT
C         **    LINEAR X AXIS NUMERICS  **                               SLPLOT
      DX = XL/FLOAT(NX)                                                  SLPLOT
      EX = X0                                                            SLPLOT
      DD = (XN-X0)/FLOAT(NX)                                             SLPLOT
      ND = 0.99 - ALOG10(ABS(DD))                                        SLPLOT
      IF  (ND.LE.0)  ND = -1                                             SLPLOT
      IX = 0                                                             SLPLOT
      YC = YP - 2.0*HN                                                   SLPLOT
   11 AX = ABS(EX)                                                       SLPLOT
      NF = 0                                                             SLPLOT
      IF  (AX.GE.10.0)  NF = ALOG10(AX)                                  SLPLOT
      NS = 0                                                             SLPLOT
      IF  (EX.LT.0.0)   NS = 1                                           SLPLOT
      SP = NS+NF+2+ND                                                    SLPLOT
      XC = XP - 0.5*SP*WN                                                SLPLOT
      CALL NUMBER (XC, YC, HN, EX, 0.0, ND)                              SLPLOT
      XP = XP + DX                                                       SLPLOT
      EX = EX + DD                                                       SLPLOT
      IX = IX + 1                                                        SLPLOT
      IF  (ABS(EX).GT.ABS(0.1*DD))  GO TO 18                             SLPLOT
      IF  (IX.GT.NX)  GO TO 12                                           SLPLOT
      CALL PLOT (XP, YP+YL,3)                                            SLPLOT
```

```
      CALL PLOT (XP, YP   ,2)                                          SLPLOT
   18 IF  (IX.LE.NX)    GO TO 11                                       SLPLOT
   12 IF  (.NOT.NXNEG)  GO TO 14                                       SLPLOT
C         **         LOG X AXIS         **                             SLPLOT
      CALL LOGAXS (XP, YP, 0.0, -NX, XL)                               SLPLOT
      XB = XL/ALOG(10.0**(-NX))                                        SLPLOT
      XA = -XB*ALOG(X0)                                                SLPLOT
C         **     LOG X AXIS NUMERICS    **                             SLPLOT
      DX = XL/FLOAT(-NX)                                               SLPLOT
      EX = ALOG10(X0)                                                  SLPLOT
      IX = 0                                                           SLPLOT
   13 CALL NUMBER (XP-1.0*WN, YP-2.5*HN, HN, 10.0, 0.0, -1)            SLPLOT
      CALL NUMBER (XP+1.0*WN, YP-2.0*HN, HN,   EX, 0.0, -1)            SLPLOT
      XP = XP + DX                                                     SLPLOT
      EX = EX + 1.0                                                    SLPLOT
      IX = IX - 1                                                      SLPLOT
      IF  (IX.GE.NX)   GO TO 13                                        SLPLOT
   14 IF  (NXLB.LE.0)  GO TO 15                                        SLPLOT
C         **        X AXIS LABEL        **                             SLPLOT
      XPX = (XL-FLOAT(NXLB)*WL)/2.0                                    SLPLOT
      YPX = YP-4.0*HN-HL                                               SLPLOT
      CALL SYMBOL(XPX, YPX, HL, XLAB,  0.0, NXLB)                      SLPLOT
   15 IF  (NPLB1.LE.0)  GO TO 16                                       SLPLOT
C         **    PLOT LABEL - 1ST LINE   **                             SLPLOT
      XP1 = (XL-FLOAT(NPLB1)*WL)/2.0                                   SLPLOT
      YP1 = YP-4.0*HN-4.0*HL                                           SLPLOT
      CALL SYMBOL (XP1, YP1, HL, PLAB1,  0.0, NPLB1)                   SLPLOT
   16 IF  (NPLB2.LE.0)  GO TO 20                                       SLPLOT
C         **    PLOT LABEL - 2ND LINE   **                             SLPLOT
      XP2 = (XL-FLOAT(NPLB2)*WL)/2.0                                   SLPLOT
      YP2 = YP-4.0*HN-6.0*HL                                           SLPLOT
      CALL SYMBOL (XP2, YP2, HL, PLAB2,  0.0, NPLB2)                   SLPLOT
   20 XP = 0.0                                                         SLPLOT
C         **     COMPLETE AXIS GRID     **                             SLPLOT
      IF  (NYPOS)  CALL LINAXS (XL, YP,  90.0,  NY,  YL)               SLPLOT
      IF  (NYNEG)  CALL LOGAXS (XL, YP,  90.0, -NY,  YL)               SLPLOT
      IF  (NXPOS)  CALL LINAXS (XL ,YL, 180.0,  NX,  XL)               SLPLOT
      IF  (NXNEG)  CALL LOGAXS (XL ,YL, 180.0, -NX, -XL)               SLPLOT
      IF  (.NOT.NYPOS)  GO TO 22                                       SLPLOT
C         **        LINEAR Y AXIS       **                             SLPLOT
      CALL LINAXS (XP, YL, -90.0,  NY,  YL)                            SLPLOT
      YB = YL/(YN-Y0)                                                  SLPLOT
C         **    LINEAR Y AXIS NUMERICS  **                             SLPLOT
      DY = YL/FLOAT(NY)                                                SLPLOT
      EY = Y0                                                          SLPLOT
      DD = (YN-Y0)/FLOAT(NY)                                           SLPLOT
      ND = 0.99 - ALOG10(ABS(DD))                                      SLPLOT
      IF  (ND.LE.0)  ND = -1                                           SLPLOT
      IY = 0                                                           SLPLOT
      XC = XP - 1.0*HN                                                 SLPLOT
```

```
   21 AY = ABS(EY)                                                    SLPLOT
      NF = 0                                                          SLPLOT
      IF  (AY.GE.10.0)  NF = ALOG10(AY)                               SLPLOT
      NS = 0                                                          SLPLOT
      IF  (EY.LT.0.0)   NS = 1                                        SLPLOT
      SP = NS+NF+2+ND                                                 SLPLOT
      YC = YP - 0.5*SP*WN                                             SLPLOT
      CALL NUMBER (XC, YC, HN, EY, 90.0, ND)                          SLPLOT
      YP = YP + DY                                                    SLPLOT
      EY = EY + DD                                                    SLPLOT
      IY = IY + 1                                                     SLPLOT
      IF  (ABS(EY).GT.ABS(0.1*DD))  GO TO 19                          SLPLOT
      IF  (IY.GT.NY)  GO TO 22                                        SLPLOT
      CALL PLOT (XP+XL, YP, 3)                                        SLPLOT
      CALL PLOT (XP   , YP, 2)                                        SLPLOT
   19 IF  (IY.LE.NY)    GO TO 21                                      SLPLOT
   22 IF  (.NOT.NYNEG)  GO TO 24                                      SLPLOT
C        **           LOG Y AXIS          **                          SLPLOT
      CALL LOGAXS (XP, YL, -90.0, -NY, -YL)                           SLPLOT
      YB = YL/ALOG(10.0**(-NY))                                       SLPLOT
      YA = -YB*ALOG(Y0)                                               SLPLOT
C        **      LOG Y AXIS NUMERICS      **                          SLPLOT
      DY = YL/FLOAT(-NY)                                              SLPLOT
      EY = ALOG10(Y0)                                                 SLPLOT
      IY = 0                                                          SLPLOT
   23 CALL NUMBER (XP-1.0*HN, YP-1.0*WN, HN, 10.0, 90.0, -1)          SLPLOT
      CALL NUMBER (XP-1.5*HN, YP+1.0*WN, HN,   EY, 90.0, -1)          SLPLOT
      YP = YP + DY                                                    SLPLOT
      EY = EY + 1.0                                                   SLPLOT
      IY = IY - 1                                                     SLPLOT
      IF  (IY.GE.NY)   GO TO 23                                       SLPLOT
   24 IF  (NYLB.LE.0)  GO TO 25                                       SLPLOT
C        **          Y AXIS LABEL         **                          SLPLOT
      XPY = XP-4.0*HN                                                 SLPLOT
      YPY = (YL-FLOAT(NYLB)*WL)/2.0                                   SLPLOT
      CALL SYMBOL(XPY, YPY, HL, YLAB, 90.0, NYLB)                     SLPLOT
   25 CONTINUE                                                        SLPLOT
C        **        PLOT DATA ARRAYS       **                          SLPLOT
      NSYM = 24                                                       SLPLOT
      IS = NPTS/NSYM                                                  SLPLOT
      IF (IS.EQ.0) IS = 1                                             VARTTH
      XOMIN = X0/1000.0                                               SLPLOT
      YOMIN = Y0/1000.0                                               SLPLOT
      DO  40  J=1,NYY                                                 SLPLOT
      IPEN = 3                                                        SLPLOT
      DO  39  I=1,NPTS                                                SLPLOT
      X1 = X2                                                         SLPLOT
      Y1 = Y2                                                         SLPLOT
      IF  (NXPOS)  X2 = XB*(X(I)   -X0)                               SLPLOT
      IF  (NYPOS)  Y2 = YB*(Y(I,J)-Y0)                                SLPLOT
```

```
      IF  (NXNEG)  X2 = XA + XB*ALOG(AMAX1(X(I)  ,XOMIN))               SLPLOT
      IF  (NYNEG)  Y2 = YA + YB*ALOG(AMAX1(Y(I,J),YOMIN))               SLPLOT
      IF  (Y2.LT.0.0 .OR. Y2.GT.YL)  GO TO 33                           SLPLOT
      IF  (X2.LT.0.0 .OR. X2.GT.XL)  GO TO 33                           SLPLOT
      IF  (IPEN.EQ.3)  GO TO 33                                         SLPLOT
      CALL PLOT (X2,Y2,IPEN)                                            SLPLOT
C         **      PLOT NYSM SYMBOLS      **                             SLPLOT
      IF  (NYY.EQ.1 .OR. MOD(I,IS).NE.0)  GO TO 39                      SLPLOT
      IF  (MOD((I/IS)-1,NYY)+1.EQ.J)  CALL SYMBOL (X2,Y2,0.14,J,0.0,-2) SLPLOT
      GO TO 39                                                          SLPLOT
   33 IF  (I.EQ.1)     GO TO 39                                         SLPLOT
      DX  = X2 - X1                                                     SLPLOT
      IF  (DX.NE.0.0)  GO TO 34                                         SLPLOT
      AX0 = 1.0                                                         SLPLOT
      AXL = 0.0                                                         SLPLOT
      IF  (X1.GE.0.0)  AX0 = 0.0                                        SLPLOT
      IF  (X1.LE.XL )  AXL = 1.0                                        SLPLOT
      GO TO 35                                                          SLPLOT
   34 AX0 =     -X1 /DX                                                 SLPLOT
      AXL = (XL-X1)/DX                                                  SLPLOT
   35 AX1 = AMIN1(AX0,AXL)                                              SLPLOT
      AX2 = AMAX1(AX0,AXL)                                              SLPLOT
      DY  = Y2 - Y1                                                     SLPLOT
      IF  (DY.NE.0.0)  GO TO 36                                         SLPLOT
      AY0 = 1.0                                                         SLPLOT
      AYL = 0.0                                                         SLPLOT
      IF  (Y1.GE.0.0)  AY0 = 0.0                                        SLPLOT
      IF  (Y1.LE.YL )  AYL = 1.0                                        SLPLOT
      GO TO 37                                                          SLPLOT
   36 AY0 =     -Y1 /DY                                                 SLPLOT
      AYL = (YL-Y1)/DY                                                  SLPLOT
   37 AY1 = AMIN1(AY0,AYL)                                              SLPLOT
      AY2 = AMAX1(AY0,AYL)                                              SLPLOT
      A1  = AMAX1(AX1,AY1,0.0)                                          SLPLOT
      A2  = AMIN1(AX2,AY2,1.0)                                          SLPLOT
      IF  (A1.GE.A2 )  GO TO 39                                         SLPLOT
      XP = X1 + A1*DX                                                   SLPLOT
      YP = Y1 + A1*DY                                                   SLPLOT
      CALL PLOT(XP,YP,IPEN)                                             SLPLOT
      IPEN = 2                                                          SLPLOT
      XP = X1 + A2*DX                                                   SLPLOT
      YP = Y1 + A2*DY                                                   SLPLOT
      CALL PLOT(XP,YP,IPEN)                                             SLPLOT
      IF  (A2.NE.1.0)  IPEN = 3                                         SLPLOT
   39 CONTINUE                                                          SLPLOT
   40 CONTINUE                                                          SLPLOT
      RETURN                                                            SLPLOT
      END                                                               SLPLOT
```

```
      SUBROUTINE SOLVA(R,AA11,AA22,AA12)                                SOLVA
C                                                   REV III.2 08/08/84REVIII
      IMPLICIT REAL*8 (A-H,O-Z)                                         SOLVA
      DIMENSION R(2,3)                                                  SOLVA
      A11=R(1,1)**2                                                     SOLVA
      A12=2.0*R(2,1)*R(1,1)                                             SOLVA
      A13=R(2,1)**2                                                     SOLVA
      A21=R(1,2)**2                                                     SOLVA
      A22=2.0*R(2,2)*R(1,2)                                             SOLVA
      A23=R(2,2)**2                                                     SOLVA
      A31=R(1,3)**2                                                     SOLVA
      A32=2.0*R(2,3)*R(1,3)                                             SOLVA
      A33=R(2,3)**2                                                     SOLVA
      DEL=A11*(A22*A33-A23*A32)-A12*(A21*A33-A23*A31)+                  SOLVA
     *    A13*(A21*A32-A22*A31)                                         SOLVA
      AA11=((A22-A12)*(A33-A23)-(A23-A13)*(A32-A22))/DEL                SOLVA
      AA12=((A23-A13)*(A31-A21)-(A21-A11)*(A33-A23))/DEL                SOLVA
      AA22=((A21-A11)*(A32-A22)-(A22-A12)*(A31-A21))/DEL                SOLVA
      RETURN                                                            SOLVA
      END                                                               SOLVA
```

```
      SUBROUTINE SOLVR(A1,A2,A3,A4,A5,A6,A7,A8,P,RX,RZ)                 SOLVR
C                                                        REV III.2 08/08/84REVIII
      IMPLICIT REAL*8 (A-H,O-Z)                                         SOLVR
C                                                                       SOLVR
C        *****************************************                      SOLVR
C                                                                       SOLVR
C        THIS SUBROUTINE WILL SOLVE A SET OF SIMULTANEOUS EQUATIONS     SOLVR
C       TO FIND COMPONETS OF VECTOR R THAT SATISFY THE PROPERTIES NEEDEDSOLVR
C      TO DETURMINE THE EQUATION OF THE PROJECTED ELLIPSE.              SOLVR
C                                                                       SOLVR
C            SEE WRITEUP.                                               SOLVR
C                                                                       SOLVR
C        *****************************************                      SOLVR
      DIMENSION P(3)                                                    SOLVR
      B=A1*P(1)+A2*P(2)+A3*P(3)                                         SOLVR
      D=A4*P(1)+A5*P(2)+A6*P(3)                                         SOLVR
      T1=A7*(D/B)**2+A6-2.0*A8*D/B                                      SOLVR
      T2=2.0*A7*D/(B)**2-2.0*A8/B                                       SOLVR
      T3=A7*(1/B)**2-1                                                  SOLVR
      RZ=(-T2+DSQRT(T2**2-4.0*T1*T3))/(2.0*T1)                          SOLVR
      RX=-D*RZ/B-1.0/B                                                  SOLVR
      RETURN                                                            SOLVR
      END                                                               SOLVR
```

```
      SUBROUTINE SPDAMP                                                  SPDAMP
C                                                  REV IV   07/24/86SLIP
C     COMPUTES THE SPRING AND VISCOUS FORCE OF A SPRING DAMPER BETWEEN   SPDAMP
C     SPECIFIED POINTS ON SELECTED SEGMENTS AND ADDS THE RESULTING       SPDAMP
C     FORCE AND TORQUE TO THE U1 AND U2 ARRAYS.                          SPDAMP
C                                                                        SPDAMP
      IMPLICIT REAL*8(A-H,O-Z)                                           SPDAMP
      COMMON/CONTRL/ TIME,NSEG,NJNT,NPL,NBLT,NBAG,NVEH,NGRND,            SPDAMP
     *               NS,NQ,NSD,NFLX,NHRNSS,NWINDF,NJNTF,NPRT(36),NPG     PAGE
      COMMON/SGMNTS/ D(3,3,30),WMEG(3,30),WMEGD(3,30),U1(3,30),U2(3,30),SPDAMP
     *               SEGLP(3,30),SEGLV(3,30),SEGLA(3,30),NSYM(30)        SPDAMP
      COMMON/DAMPER/ APSDM(3,20),APSDN(3,20),ASD(5,20),MSDM(20),MSDN(20)SPDAMP
      COMMON/TABLES/MXNTI,MXNTB,MXTB1,MXTB2,NTI(50),NTAB(1250),TAB(4500)BUTLER2
      COMMON/FORCES/PSF(7,70),BSF(4,20),SSF(10,40),BAGSF(3,20),          NCFORC
     *               PRJNT(7,30),NPANEL(5),NPSF,NBSF,NSSF,NBGSF          SPDAMP
      COMMON/TEMPVS/DELM(3),DELN(3),DD(3),DEL,T1(3),T2(3),T3(3),T4(3),   SPDAMP
     *             DUNIT(3),DV(3),DMV,DDO,FS,FD,TOTF(3),                 SPDAMP
     *             T5(3),T6(3),T7(3),T8(3)                               SPDAMP
      CALL ELTIME(1,32)                                                  SPDAMP
      NBSFO = NBSF                                                       SPDAMP
      DO 90 I=1,NSD                                                      SPDAMP
      M = MSDM(I)                                                        SPDAMP
      N = MSDN(I)                                                        SPDAMP
C                                                                        SPDAMP
C     COMPUTE VECTOR AND ITS MAGNITUDE BETWEEN THE SPECIFIED POINTS.     SPDAMP
C                                                                        SPDAMP
      CALL DOT31 (D(1,1,M),APSDM(1,I),DELM)                              SPDAMP
      CALL DOT31 (D(1,1,N),APSDN(1,I),DELN)                              SPDAMP
      DEL = 0.0                                                          SPDAMP
      DO 10 K=1,3                                                        SPDAMP
      DD(K) = SEGLP(K,M)+DELM(K)-SEGLP(K,N)-DELN(K)                      SPDAMP
   10 DEL = DEL+DD(K)**2                                                 SPDAMP
      IF  (DEL.LE.0.0)  GO TO 90                                         SPDAMP
      DEL = DSQRT(DEL)                                                   SPDAMP
C                                                                        SPDAMP
C     COMPUTE RELATIVE VELOCITY AND ITS COMPONENT ON VECTOR LINE.        SPDAMP
C                                                                        SPDAMP
      CALL CROSS(WMEG(1,M),APSDM(1,I),T1)                                SPDAMP
      CALL CROSS(WMEG(1,N),APSDN(1,I),T2)                                SPDAMP
      CALL DOT31 (D(1,1,M),T1,T3)                                        SPDAMP
      CALL DOT31 (D(1,1,N),T2,T4)                                        SPDAMP
      DO 20 K=1,3                                                        SPDAMP
      DUNIT(K) = DD(K)/DEL                                               SPDAMP
   20 DV(K) = SEGLV(K,M)+T3(K)-SEGLV(K,N)-T4(K)                          SPDAMP
      DMV = DUNIT(1)*DV(1)+DUNIT(2)*DV(2)+DUNIT(3)*DV(3)                 SPDAMP
C                                                                        SPDAMP
C     COMPUTE SPRING AND VISCOUS FORCE AND THE COMPONENTS                SPDAMP
C     ALONG THE UNIT VECTOR                                              SPDAMP
C                                                                        SPDAMP
      FS = 0.0                                                           SPDAMP
```

```
      FD = 0.0                                                          SPDAMP
      IF  (ASD(1,I).LT.0.0)  GO TO 21                                   SLIP
      DD0 = DEL-ASD(1,I)                                                SPDAMP
      IF  (DD0.LE.0.0 .AND. ASD(2,I).LE.0.0)  GO TO 41                  SPDAMP
      FS = DD0*(DABS(ASD(2,I)) + DABS(DD0)*ASD(3,I))                    SPDAMP
      FD = DMV*(ASD(4,I)+DABS(DMV)*ASD(5,I))                            SPDAMP
      GO TO 29                                                          SPDAMP
   21 DD0 = DEL+ASD(1,I)                                                SPDAMP
      JF1 = ASD(2,I)                                                    SPDAMP
      IF  (JF1.EQ.C)  GO TO 22                                          SPDAMP
      JF2 = NTI(JF1)                                                    SPDAMP
      IF  (DD0.GT.0.0 .OR. ASD(3,I).EQ.0.0)  FS = EVALFD(DD0,JF2,1)     SPDAMP
   22 JF3 = ASD(4,I)                                                    SPDAMP
      IF  (JF3.EQ.0)  GO TO 29                                          SPDAMP
      JF4 = NTI(JF3)                                                    SPDAMP
      IF  (DD0.GT.0.0 .OR. ASD(3,I).EQ.0.0)  FD = EVALFD(DMV,JF4,1)     SLIP
   29 DO 30 K=1,3                                                       SPDAMP
   30 TOTF(K) = (FS+FD)*DUNIT(K)                                        SPDAMP
C                                                                       SPDAMP
C     AND ADD THE RESULTING FORCE AND TORQUE TO THE U1 AND U2 ARRAYS.   SPDAMP
C                                                                       SPDAMP
      CALL MAT31(D(1,1,M),TOTF,T5)                                      SPDAMP
      CALL MAT31(D(1,1,N),TOTF,T6)                                      SPDAMP
      CALL CROSS(APSDM(1,I),T5,T7)                                      SPDAMP
      CALL CROSS(APSDN(1,I),T6,T8)                                      SPDAMP
      DO 40 K=1,3                                                       SPDAMP
      U1(K,M) = U1(K,M) - TOTF(K)                                       SPDAMP
      U1(K,N) = U1(K,N) + TOTF(K)                                       SPDAMP
      U2(K,M) = U2(K,M) - T7(K)                                         SPDAMP
   40 U2(K,N) = U2(K,N) + T8(K)                                         SPDAMP
   41 IBSF = 3-2*MOD(I,2)                                               SPDAMP
      NBSF = NBSF0 + (I+1)/2                                            SPDAMP
      BSF(IBSF  ,NBSF) = DEL                                            SPDAMP
      BSF(IBSF+1,NBSF) = FD + FS                                        SPDAMP
   90 CONTINUE                                                          SPDAMP
      CALL ELTIME(2,32)                                                 SPDAMP
      RETURN                                                            SPDAMP
      END                                                               SPDAMP
```

```
      SUBROUTINE SPLINE (X,Y,F,N,L)                                     SPLINE
C                                                  REV 19    05/14/79SPLINE
C                                                                       SPLINE
C     ROUTINE TO FIT A SET OF POLYNOMIALS OF DEGREE L                   SPLINE
C     TO A SET OF GIVEN DATA POINTS (X(I),Y(I),I=1,N)                   SPLINE
C                                                                       SPLINE
C     FUNCTION IS OF FORM:                                              SPLINE
C                                                                       SPLINE
C       Y = F(2,K) + F(3,K)*DX + F(4,K)*DX**2 + F(5,K)*DX**3            SPLINE
C                                                                       SPLINE
C       WHERE: DX = XX - F(1,K)                                         SPLINE
C              F(1,K) .LE. XX .LT. F(1,K+1)  ;  (SETS K)                SPLINE
C              IF  (XX.GT.F(1,N))  ;  USE K=N, CONSTANT FIT TO Y(N)     SPLINE
C              IF  (XX.LT.F(1,1))  ;  EXTRAPOLATED FIT FOR K=1          SPLINE
C                                                                       SPLINE
C                 F(1,I) = X(I) ,                I=1,N                  SPLINE
C                 F(2,I) = Y(I) ,                I=1,N                  SPLINE
C                                                                       SPLINE
C      DEGREE L                                         CONTINUITY      SPLINE
C             0   F(3,I) = F(4,I) = F(5,I) = 0 , I=1,N     NONE         SPLINE
C             1   F(4,I) = F(5,I) = 0  ,         I=1,N     Y            SPLINE
C             2   F(5,I) = 0 ,                   I=1,N     Y,Y'         SPLINE
C             3   CUBIC SPLINE                             Y,Y',Y''     SPLINE
C                                                                       SPLINE
C                 F(K,N)=0 FOR K=3,5 IN ALL CASES                       SPLINE
C                                                                       SPLINE
C     FOR L=2 AND L=3 THE CHANGES IN THE L'TH DERIVATIVES ARE MINIMIZED SPLINE
C                                                                       SPLINE
C     SPECIAL CASES:                                                    SPLINE
C         N=1 ;      TREATED AS L=0                                     SPLINE
C         N=2 ;      TREATED AS L=MIN(L,1)                              SPLINE
C         L<0 ;      TREATED AS L=0                                     SPLINE
C         L>3 ;      TREATED AS L=3                                     SPLINE
C                                                                       SPLINE
C     STORAGE REQUIRED  X(N),Y(N),F(5,N); SET BY CALLING PROGRAM        SPLINE
C                                                                       SPLINE
C     USAGE:                                                            SPLINE
C       ALL COMPUTATIONS AND REAL VARIABLES ARE DOUBLE PRECISION        SPLINE
C       GIVEN:  L,N, (X(I),Y(I),I=1,N)                                  SPLINE
C       CALL SPLINE (X,Y,F,N,L)         ; SETS F                        SPLINE
C                                                                       SPLINE
CC                                                                      SPLINE
CC      TO EVALUATE FUNCTION AND DERIVATIVES AT POINT XX                SPLINE
CC                                                                      SPLINE
C           DO  10  K=1,N                                               SPLINE
C           IF (K.EQ.N)  GO TO 11                                       SPLINE
C           IF (XX.LT.F(1,K+1)) GO TO 11                                SPLINE
C       10  CONTINUE                                                    SPLINE
C       11  DX = XX - F(1,K)                                            SPLINE
C           YY = F(2,K) + DX*(F(3,K)+DX*(F(4,K)+DX*F(5,K)))             SPLINE
```

```
C           YD = F(3,K) + DX*(2.0*F(4,K)+3.0*DX*F(5,K))                     SPLINE
C           YDD = 2.0*F(4,K) + 6.0*DX*F(5,K)                                SPLINE
C           YDDD = 6.0*F(5,K)                                               SPLINE
C           YDDDD = 0.0                                                     SPLINE
CC                                                                          SPLINE
CC        FUNCTIONAL VALUE IN YY, DERIVATIVES IN YD'S                       SPLINE
CC        REPEAT FOR NEXT VALUE OF XX                                       SPLINE
C                                                                           SPLINE
C     AUTHOR: DR. JOHN T. FLECK                                             SPLINE
C                                                                           SPLINE
      IMPLICIT REAL*8 (A-H,O-Z)                                             SPLINE
      DIMENSION X(N),Y(N),F(5,N),C(2,3)                                     SPLINE
      DO  20  I=1,N                                                         SPLINE
      F(1,I) = X(I)                                                         SPLINE
      DO  10  K=2,5                                                         SPLINE
   10 F(K,I) = 0.0                                                          SPLINE
      IF  (L.LT.3)  F(2,I) = Y(I)                                           SPLINE
   20 IF  (L.GT.0 .AND. I.LT.N)  F(3,I) = (Y(I+1)-Y(I))/(X(I+1)-X(I))       SPLINE
      IF  (L.LT.2 .OR.  N.LT.3)  GO TO 99                                   SPLINE
      IF  (L.GE.3)  GO TO 50                                                SPLINE
      D1 = X(2) - X(1)                                                      SPLINE
      SS = 0.0                                                              SPLINE
      DS = 0.0                                                              SPLINE
      DO  30  I=3,N                                                         SPLINE
      F(4,I-1) = F(3,I-1) - F(3,I-2) - F(4,I-2)                             SPLINE
      DX1 = X(I) - X(I-1)                                                   SPLINE
      DX2 = X(I-1) - X(I-2)                                                 SPLINE
      DD = D1/DX1 + D1/DX2                                                  SPLINE
      SS = SS + DD*DD                                                       SPLINE
      DS = DS + DD*(F(4,I-1)/DX1 - F(4,I-2)/DX2)                            SPLINE
   30 D1 = -D1                                                              SPLINE
      F(4,1) = DS/SS                                                        SPLINE
      DX = (X(2)-X(1))*F(4,1)                                               SPLINE
      F(3,1) = F(3,1) - DX                                                  SPLINE
      DO  40  I=3,N                                                         SPLINE
      XX = F(4,I-1) - DX                                                    SPLINE
      F(3,I-1) = F(3,I-1) - XX                                              SPLINE
      F(4,I-1) = XX/(X(I)-X(I-1))                                           SPLINE
   40 DX = -DX                                                              SPLINE
      GO TO 99                                                              SPLINE
C                                                                           SPLINE
C     CUBIC SPLINE                                                          SPLINE
C                                                                           SPLINE
   50 DO  51  I=2,N                                                         SPLINE
      IF  (I.EQ.N)  GO TO 51                                                SPLINE
      F(4,I) = 3.0*(F(3,I)-F(3,I-1))                                        SPLINE
      F(5,I) = 2.0*(X(I+1)-X(I-1))                                          SPLINE
   51 F(3,I-1) = 0.0                                                        SPLINE
      F(2,N) = -1.0                                                         SPLINE
      F(3,1) = -1.0                                                         SPLINE
```

```
      DO  60  I=3,N                                               SPLINE
      DX = X(I-1) - X(I-2)                                        SPLINE
      IF  (I.GT.3)  DX = DX/F(5,I-2)                              SPLINE
      DO  60  K=3,5                                               SPLINE
   60 F(K,I-1) = F(K,I-1)- F(K,I-2)*DX**((K-1)/2)                 SPLINE
      DO  70  I=3,N                                               SPLINE
      NI = N-I                                                    SPLINE
      DX = X(NI+3) - X(NI+2)                                      SPLINE
      DO  70  K=2,4                                               SPLINE
   70 F(K,NI+2) = (F(K,NI+2) - DX*F(K,NI+3))/F(5,NI+2)            SPLINE
      DO  71  J=1,2                                               SPLINE
      DO  71  K=J,3                                               SPLINE
      C(J,K) = 0.0                                                SPLINE
      DO  71  I=3,N                                               SPLINE
      DX1 = X(I) - X(I-1)                                         SPLINE
      DX2 = X(I-1) - X(I-2)                                       SPLINE
   71 C(J,K) = C(J,K) + (  (F(J+1,I  )-F(J+1,I-1))/DX1            SPLINE
     *                   - (F(J+1,I-1)-F(J+1,I-2))/DX2)           SPLINE
     *                 * (  (F(K+1,I  )-F(K+1,I-1))/DX1           SPLINE
     *                   - (F(K+1,I-1)-F(K+1,I-2))/DX2)           SPLINE
      DEN    =  C(1,1)*C(2,2) - C(1,2)*C(1,2)                     SPLINE
      F(4,1) = (C(1,1)*C(2,3) - C(1,2)*C(1,3))/DEN                SPLINE
      F(4,N) = (C(2,2)*C(1,3) - C(1,2)*C(2,3))/DEN                SPLINE
      DO  72  I=3 N                                               SPLINE
   72 F(4,I-1) = F(4,I-1) - F(4,1)*F(3,I-1) - F(4,N)*F(2,I-1)     SPLINE
      D1 = X(2) - X(1)                                            SPLINE
      F(3,1) = (Y(2)-Y(1))/D1 - (2.0*F(4,1)+F(4,2))*D1/3.0        SPLINE
      F(2,1) = Y(1)                                               SPLINE
      DO  80  I=2,N                                               SPLINE
      F(2,I) = Y(I)                                               SPLINE
      DX = X(I) - X(I-1)                                          SPLINE
      IF  (I.LT.N)  F(3,I) = F(3,I-1) + (F(4,I)+F(4,I-1))*DX      SPLINE
   80 F(5,I-1) = (F(4,I)-F(4,I-1))/(3.0*DX)                       SPLINE
      F(4,N) = 0.0                                                SPLINE
   99 RETURN                                                      SPLINE
      END                                                         SPLINE
```

```
      DOUBLE PRECISION FUNCTION SPRNGF(T,D,ZD,SPR,JSTOP)               SPRNGF
C                                                    REV IV    07/23/86TWOPI
C     COMPUTES NONLINEAR SPRING TORQUE FOR JOINTS AS A FUNCTION OF ANGLESPRNGF
C     ACTUALLY ROUTINE RETURNS TORQUE/ABS(SIN THETA)                   SPRNGF
C                                                                      SPRNGF
C     ARGUMENTS:                                                       SPRNGF
C          T     : COS THETA WHERE THETA IS ANGLE OF JOINT (0<THETA<PI) SPRNGF
C          D     : ABS(SIN THETA)                                      SPRNGF
C          ZD    : -THETA DOT * SIN THETA                              SPRNGF
C          SPR   : ARRAY OF 5 VALUES DESCRIBING FUNCTION EVALUATION    SPRNGF
C          JSTOP : INDICATOR TO BE SET TO ONE IF JOINT IS IN STOP      SPRNGF
C                                                                      SPRNGF
      IMPLICIT REAL*8 (A-H,O-Z)                                        SPRNGF
      COMMON/CNSNTS/ PI,RADIAN,G,THIRD,EPS(24),                        SPRNGF
     *               UNITL,UNITM,UNITT,GRAVTY(3),TWOPI                 TWOPI
      DIMENSION SPR(5)                                                 SPRNGF
C                                                                      SPRNGF
C      RESET T=1 IF T>1 (HAD & HBD IN VISPR)                           SPRNGF
C                                                                      SPRNGF
      IF (T.GT.1.0) T = 1.0                                            SPRNGF
      IF (T.LT.-1.0) T = -1.0                                          SPRNGF
      Z = DACOS(T)                                                     SPRNGF
      U = EPS(1)*D                                                     SPRNGF
      Q = 0.0                                                          SPRNGF
      IF (D.NE.0.0) Q = -ZD/U                                          SPRNGF
      IF (Q.GT.1.0) Q = 1.0                                            SPRNGF
      IF (Q.LT.-1.0) Q = -1.0                                          SPRNGF
      X = 0.5*(1.0+SPR(4) + Q*(1.0-SPR(4)) )                           SPRNGF
      Y = 0.0                                                          SPRNGF
      IF (D.NE.0.0) Y = Z/D                                            SPRNGF
      Q = 1.0                                                          SPRNGF
      IF (DABS(Z).LT.EPS(4)) Y = DSIGN(Q,Z)                            SPRNGF
      SPRNGF = Y*SPR(1)                                                SPRNGF
      JSTOP = 0                                                        SPRNGF
      IF (SPR(5).GT.0.0) GO TO 10                                      SPRNGF
      SPRNGF = X*SPRNGF                                                SPRNGF
      GO TO 11                                                         SPRNGF
   10 IF (Z.LT.SPR(5)) GO TO 11                                        SPRNGF
      JSTOP = 1                                                        SPRNGF
      Z = Z-SPR(5)                                                     SPRNGF
      SPRNGF = SPRNGF + X/D*(SPR(2)+Z*SPR(3))*Z**2                     SPRNGF
   11 CONTINUE                                                         SPRNGF
      RETURN                                                           SPRNGF
      END                                                              SPRNGF
```

```
      SUBROUTINE TRIGFS                                                 TRIGFS
C                                                 REV 19    08/05/78TRIGFS
      IMPLICIT REAL*8 (A-H,O-Z)                                         TRIGFS
      COMMON/CDINT/ UU(4),GH(3,4),                                      TRIGFS
     *              E(3,240), F(5,240),GG(5,240),Y(5,240),U(5,240),     TRIGFS
     *              H,HPRINT,HS,TPRINT,TSTART,ICNT,IDBL,IFLAG           TRIGFS
      BETA = 0.0                                                        TRIGFS
      IF (HS.NE.0.0) BETA = (H/HS)**2                                   TRIGFS
      R1 = HS/H                                                         TRIGFS
      R2 = 1.0+BETA*R1                                                  TRIGFS
      GH(3,1) = 2.0/(H*R2)                                              TRIGFS
      GH(2,1) = GH(3,1)*(BETA-1.0)                                      TRIGFS
      GH(1,1) = GH(3,1)* BETA                                           TRIGFS
      GH(1,2) = 4.0*BETA/(R2*H**2)                                      TRIGFS
      GH(3,2) = GH(1,2)* R1                                             TRIGFS
      GH(2,2) = GH(1,2)*(R1+1.0)                                        TRIGFS
      GH(3,3) = 1.0/H                                                   TRIGFS
      GH(2,3) = 4.0*GH(3,3)                                             TRIGFS
      GH(1,3) = 3.0*GH(3,3)                                             TRIGFS
      GH(3,4) = 2.0/H**2                                                TRIGFS
      GH(2,4) = 2.0*GH(3,4)                                             TRIGFS
      GH(1,4) =     GH(3,4)                                             TRIGFS
      UU(1) = 2.0/H                                                     TRIGFS
      UU(2) = 0.0                                                       TRIGFS
      UU(3) = 0.5*H                                                     TRIGFS
      UU(4) = 0.25*H**2                                                 TRIGFS
      IF  (HS.EQ.0.0)  GO TO 99                                         TRIGFS
      UU(1) = BETA*(4.25+2.25/R1)                                       TRIGFS
      UU(2) = BETA*(2.25+1.25/R1)/R1                                    TRIGFS
      UAU   = 1.0+UU(1)+UU(2)                                           TRIGFS
      UU(1) = 2.0*UU(1)/(UAU*H)                                         TRIGFS
      UU(2) = 4.0*UU(2)/(UAU*H**2)                                      TRIGFS
   99 RETURN                                                            TRIGFS
      END                                                               TRIGFS
```

```
      SUBROUTINE UNIT1(IND)                                              UNIT1
C                                                  REV IV    02/20/87HYPER
C     THIS SUBROUTINE REPLACES THE PROGRAM CODE THAT PREVIOUSLY WAS      UNIT1
C     NEAR THE END OF THE MAIN PROGRAM TO WRITE ON UNIT 1 THAT DATA      UNIT1
C     USED FOR VARIOUS PLOTTING PROGRAMS (E.G. BUBBLE MAN PLOT).         UNIT1
C                                                                        UNIT1
C     THIS SUBROUTINE IS WRITTEN TO GENERATE UNIT 1 IN SUCH A MANNER     UNIT1
C     TO BE COMPATIBLE WITH THE INPUT REQUIREMENTS FOR THE AAMRL VIEW    UNIT1
C     PROGRAM THAT IS NOW BEING DISTRIBUTED ON THE CVS PROGRAM TAPES.    UNIT1
C                                                                        UNIT1
C     ARGUMENTS:                                                         UNIT1
C          IND = 0: CALL IS FROM THE MAIN PROGRAM                        UNIT1
C              * 0: CALL IS FROM SUBROUTINE EQUILB                       UNIT1
C                                                                        UNIT1
      IMPLICIT REAL*8 (A-H,O-Z)                                          UNIT1
      COMMON/CONTRL/ TIME,NSEG,NJNT,NPL,NBLT,NBAG,NVEH,NGRND,            UNIT1
     *               NS,NQ,NSD,NFLX,NHRNSS,NWINDF,NJNTF,NPRT(36),NPG     PAGE
      COMMON/SGMNTS/ D(3,3,30),WMEG(3,30),WMEGD(3,30),U1(3,30),U2(3,30),UNIT1
     *               SEGLP(3,30),SEGLV(3,30),SEGLA(3,30),NSYM(30)        UNIT1
      COMMON/CNTSRF/ PL(24,30),BELT(20,8),TPTS(6,8),BD(24,40)            EDGE
      COMMON/JBARTZ/ MNPL(   30),MNBLT(   8),MNSEG(   30),MNBAG(    6),  UNIT1
     *               MPL(3,5,30),MBLT(3,5,8),MSEG(3,5,30),MBAG(3,10,6),  UNIT1
     *               NTPL( 5,30),NTBLT( 5,8),NTSEG( 5,30)                UNIT1
      COMMON/RSAVE/  XSG(3,20,3),DPMI(3,3,30),LPMI(30),                  UNIT1
     *               NSG(9),MSG(20,9),MCG,MCGIN(24,5),KREF(20,9)         TTHKREF
      COMMON/TEMPVS/ XD(3,3,30),XSEGLP(3,30),XPL(17,30),XBD(24,40),      UNIT1
     *               T1(3),T3(3,3)                                       FIXWBS
      REAL XTIME,XD,XSEGLP,XPL,XBD                                       UNIT1
      DATA IFIRST/0/                                                     UNIT1
      IF  (NPRT(1).EQ.0)  GO TO 99                                       UNIT1
      IF  (IFIRST.NE.0)  GO TO 20                                        UNIT1
      IFIRST = 1                                                         UNIT1
C                                                                        UNIT1
C     FIRST TIME IN ROUTINE, WRITE STATIC DATA ON OUTPUT UNIT 1.         UNIT1
C     DATA MUST BE CONVERTED TO SINGLE PRECISION FOR VIEW PROGRAM.       UNIT1
C                                                                        UNIT1
      DO  11  J=1,30                                                     UNIT1
      DO 11 I=1,17                                                       FIXWBS
   11 XPL(I,J) = PL(I,J)                                                 FIXWBS
      DO  12  J=1,40                                                     UNIT1
      K = 1                                                              HYPER
      IF (BD(1,J).LT.0.0) K = 2                                          HYPER
      DO  12  I=1,24                                                     UNIT1
      XBD(I,J) = BD(K,J)                                                 HYPER
   12 K = K + 1                                                          HYPER
      WRITE (1) NSEG,NPL,XPL,XBD,MPL                                     UNIT1
      GOTO 99                                                            EDGE
C                                                                        UNIT1
C     WRITE TIME POINT DATA ON OUTPUT UNIT 1.                            UNIT1
C     DATA MUST BE CONVERTED TO SINGLE PRECISION FOR VIEW PROGRAM.       UNIT1
```

```
C                                                                       UNIT1
   20 XTIME = TIME                                                      UNIT1
      DO  22  K=1,30                                                    UNIT1
      DO  22  J=1,3                                                     UNIT1
      DO  21  I=1,3                                                     UNIT1
   21 XD(I,J,K) = D(I,J,K)                                              UNIT1
   22 XSEGLP(J,K) = SEGLP(J,K)                                          UNIT1
      DO  25  K=1,NSEG                                                  UNIT1
      IF  (LPMI(K).EQ.0)  GO TO 25                                      UNIT1
      CALL DOT33 (DPMI(1,1,K),D(1,1,K),T3)                              UNIT1
      DO  24  I=1,3                                                     UNIT1
      DO  24  J=1,3                                                     UNIT1
   24 XD(I,J,K) = T3(I,J)                                               UNIT1
   25 CONTINUE                                                          UNIT1
      WRITE (1) XTIME,XSEGLP,XD                                         UNIT1
   99 RETURN                                                            UNIT1
      END                                                               UNIT1
```

```
      SUBROUTINE UPDATE(I)                                              UPDATE
C                                                  REV IV    07/24/86SLIP
C     CALLED BY SUBROUTINE DINT                                         UPDATE
C                                                                       UPDATE
C       (I=1)  AT THE START OF A NEW STEP TO SETUP ANY NEW CONDITIONS   UPDATE
C              TO BE VALID FOR ENTIRE INTEGRATION STEP                  UPDATE
C                A. UPDATE FORCE DEFLECTION FUNCTIONS(SUBROUTINE UPDFDC)UPDATE
C                B. TEST FOR LOCKED JOINTS                              UPDATE
C       NOTE: ARGUMENT I WILL BE SET TO -1 TO RESET INTEGRATOR.         UPDATE
C                                                                       UPDATE
C       (I=2)  AT THE END OF EACH SUCCESSFUL INTEGRATION STEP TO        UPDATE
C              COMPLETE CALCULATIONS FOR OUTPUT (SUBROUTINE AIRBG3).    UPDATE
C                                                                       UPDATE
      IMPLICIT REAL*8(A-H,O-Z)                                          UPDATE
      COMMON/CONTRL/ TIME,NSEG,NJNT,NPL,NBLT,NBAG,NVEH,NGRND,           UPDATE
     *               NS,NQ,NSD,NFLX,NHRNSS,NWINDF,NJNTF,NPRT(36),NPG    PAGE
      COMMON/SGMNTS/ D(3,3,30),WMEG(3,30),WMEGD(3,30),U1(3,30),U2(3,30),UPDATE
     *               SEGLP(3,30),SEGLV(3,30),SEGLA(3,30),NSYM(30)       UPDATE
      COMMON/DESCRP/ PHI(3,30),W(30),RW(30),SR(4,60),HA(3,60),HB(3,60), SLIP
     *               RPHI(3,30),HT(3,3,60),SPRING(5,90),VISC(7,90),     UPDATE
     *               JNT(30),IPIN(30),ISING(30),IGLOB(30),JOINTF(30)    UPDATE
      COMMON/CMATRX/ V1(3,30),V2(3,30),V3(3,12),B12(3,3,60),A22(3,3,60),UPDATE
     *               F(3,30),TQ(3,30),WJ(30),A11(3,3,30)                SLIP
      COMMON/JBARTZ/ MNPL(   30),MNBLT(   8),MNSEG(   30),MNBAG(    6), UPDATE
     *               MPL(3,5,30),MBLT(3,5,8),MSEG(3,5,30),MBAG(3,10,6), UPDATE
     *               NTPL( 5,30),NTBLT( 5,8),NTSEG( 5,30)               UPDATE
      COMMON/TABLES/MXNTI,MXNTB,MXTB1,MXTB2,NTI(50),NTAB(1250),TAB(4500)UPDATE
      COMMON/FORCES/PSF(7,70),BSF(4,20),SSF(10,40),BAGSF(3,20),         NCFORC
     *               PRJNT(7,30),NPANEL(5),NPSF,NBSF,NSSF,NBGSF         UPDATE
      COMMON/CSTRNT/ A13(3,3,24),A23(3,3,24),B31(3,3,24),B32(3,3,24),   UPDATE
     *               HHT(3,3,12),RK1(3,12),RK2(3,12),QQ(3,12),TQQ(3,12),UPDATE
     *               RQQ(3,12),HQQ(3,12),SQQ(12),CFQQ(12),              UPDATE
     *               KQ1(12),KQ2(12),KQTYPE(12)                         UPDATE
      COMMON/TEMPVI/ CREST,TTI(3),R1I(3),R2I(3),JSTOP(4,2,30)           UPDATE
      COMMON/CEULER/ IEULER(30),HIR(3,3,90),ANG(3,30),ANGD(3,30),       JDRIFT
     *               FE(3,30),TQE(3,30),CONST(5,30)                     JDRIFT
      COMMON/HRNESS/ BAR(15,100),BB(100),BBDOT(100),PLOSS(2,100),       UPDATE
     *               XLONG(20),HTIME(2),IBAR(5,100),NL(2,100),          UPDATE
     *               NPTSPB(20),NPTPLY(20),NTHRNS(20),NBLTPH(5)         UPDATE
      DIMENSION TQTEST(3),LOCK(8,3) ,T(3)                               UPDATE
      DATA LOCK/-8, 6, 5, 7,-3,-2,-4, 1,                                UPDATE
     *           6,-8, 4,-3, 7,-1,-5, 2,                                UPDATE
     *           5, 4,-8,-2,-1, 7,-6, 3/                                UPDATE
C                                                                       UPDATE
C     CALL AIRBG3 FOR AIRBAG, IF ANY.                                   UPDATE
C                                                                       UPDATE
      IF  (NBAG.NE.0)  CALL AIRBG3(I)                                   UPDATE
      IF  (I.EQ.2)  GO TO 42                                            UPDATE
      CALL ELTIME (1,7)                                                 UPDATE
      IF  (NPL.LE.0)  GO TO 13                                          UPDATE
```

```
C                                                                 UPDATE
C     CALL UPDFDC FOR EACH ALLOWED PLANE-SEGMENT CONTACT.         UPDATE
C                                                                 UPDATE
      NPSF = 0                                                    UPDATE
      DO 12 J=1,NPL                                               UPDATE
      NK = MNPL(J)                                                UPDATE
      IF (NK.LE.0) GO TO 12                                       UPDATE
      DO 11 K = 1, NK                                             UPDATE
      NPSF = NPSF+1                                               UPDATE
      NT = NTPL(K,J)                                              UPDATE
      NF = NTAB(NT+5)                                             UPDATE
      CALL UPDFDC(NT)                                             UPDATE
      IF (NT.GT.0.OR.TAB(NF+3).EQ.0.0) GO TO 11                   UPDATE
      CALL IMPULS(1,K,J)                                          UPDATE
      I = -1                                                      UPDATE
   11 CONTINUE                                                    UPDATE
   12 CONTINUE                                                    UPDATE
   13 IF (NBLT.LE.0) GO TO 16                                     UPDATE
C                                                                 UPDATE
C     CALL UPDFDC FOR EACH ALLOWED BELT-SEGMENT CONTACT.          UPDATE
C                                                                 UPDATE
      DO 15 J=1,NBLT                                              UPDATE
      NK = MNBLT(J)                                               UPDATE
      IF (NK.LE.0) GO TO 15                                       UPDATE
      DO 14 K = 1,NK                                              UPDATE
      NT = NTBLT(K,J)                                             UPDATE
      NF = NTAB(NT+5)                                             UPDATE
      NT6 = NT+6                                                  UPDATE
      CALL UPDFDC(NT)                                             UPDATE
C                                                                 UPDATE
C        AND FOR 2ND FUNCTION, IF FULL BELT FRICTION.             UPDATE
C                                                                 UPDATE
   14 IF (NF.NE.0) CALL UPDFDC(NT6)                               UPDATE
   15 CONTINUE                                                    UPDATE
C                                                                 UPDATE
C     CALL UPDFDC FOR EACH ALLOWED SEGMENT-SEGMENT CONTACT.        UPDATE
C                                                                 UPDATE
   16 NSSF = 0                                                    UPDATE
      DO 18 J=1,NSEG                                              UPDATE
      NK = MNSEG(J)                                               UPDATE
      IF (NK.LE.0) GO TO 18                                       UPDATE
      DO 17 K = 1,NK                                              UPDATE
      NSSF = NSSF+1                                               UPDATE
      NT = NTSEG(K,J)                                             UPDATE
      NF = NTAB(NT+5)                                             UPDATE
      CALL UPDFDC(NT)                                             UPDATE
      IF (NT.GT.0.OR.TAB(NF+3).EQ.0.0) GO TO 17                   UPDATE
      CALL IMPULS(3,K,J)                                          UPDATE
      I = -1                                                      UPDATE
   17 CONTINUE                                                    UPDATE
```

```
   18 CONTINUE                                                          UPDATE
      IF (NHRNSS.LE.0) GO TO 71                                         UPDATE
C                                                                       UPDATE
C     CALL UPDFDC FOR EACH BELT OF HARNESS-BELT SYSTEMS.                UPDATE
C                                                                       UPDATE
      CALL HPTURB                                                       UPDATE
      J1 = 1                                                            UPDATE
      K1 = 1                                                            UPDATE
      DO 70 II=1,NHRNSS                                                 UPDATE
      IF (NBLTPH(II).LE.0) GO TO 70                                     UPDATE
      J2 = J1 + NBLTPH(II) - 1                                          UPDATE
      DO 69 J=J1,J2                                                     UPDATE
      IF  (NPTPLY(J).LE.0)  GO TO 69                                    UPDATE
      NT = NTHRNS(J)                                                    UPDATE
      CALL UPDFDC(NT)                                                   UPDATE
      K2 = K1 + NPTPLY(J) - 1                                           UPDATE
      DO  68  K=K1,K2                                                   UPDATE
      KI = NL(1,K)                                                      UPDATE
      NT = IBAR(3,KI)                                                   UPDATE
      CALL UPDFDC(NT)                                                   UPDATE
   68 CONTINUE                                                          UPDATE
      K1 = K2+1                                                         UPDATE
   69 CONTINUE                                                          UPDATE
      J1 = J2+1                                                         UPDATE
   70 CONTINUE                                                          UPDATE
   71 IF (NJNT.LE.0) GO TO 37                                           UPDATE
C                                                                       UPDATE
C           CHECK FOR IMPULSE ON JOINT STOPS                            UPDATE
C     TO BE CALLED IF IN JOINT STOP (JSTOP(1)=1) THIS TIME STEP         UPDATE
C     BUT NOT IN IN JOINT STOP (JSTOP(2)=0) AT PREVIOUS TIME.           UPDATE
C                                                                       UPDATE
      DO 21 K=1,NJNT                                                    UPDATE
      IF (JNT(K).EQ.0) GO TO 21                                         UPDATE
      IF (IABS(IPIN(K)).NE.4 .AND. VISC(7,3*K-2).EQ.0.0) GO TO 20       UPDATE
      DO 19 J=1,3                                                       UPDATE
      K3J = 3*K-3+J                                                     UPDATE
      IF (IABS(IPIN(K)).NE.4) K3J=3*K-2                                 UPDATE
      IF (IABS(IPIN(K)).EQ.4 .AND. VISC(7,K3J).EQ.0.0) GO TO 19         UPDATE
      IF (JSTOP(J,1,K).NE.1.OR.JSTOP(J,2,K).NE.0) GO TO 19              UPDATE
      CALL IMPULS(4,J,K)                                                UPDATE
      I = -1                                                            UPDATE
   19 JSTOP(J,2,K) = JSTOP(J,1,K)                                       UPDATE
   20 IF (IGLOB(K).EQ.0) GO TO 21                                       UPDATE
      NT = IGLOB(K)                                                     UPDATE
      MT = NTAB(NT+5)                                                   UPDATE
      NT1 = NTAB(NT+2)                                                  UPDATE
      NTAB(NT+2) = 0                                                    UPDATE
      CALL UPDFDC(NT)                                                   UPDATE
      NT = IABS(NT)                                                     UPDATE
      NTAB(NT+2) = NT1                                                  UPDATE
```

```
      IF (TAB(MT+3).EQ.0.0) GO TO 21                                  UPDATE
      IF (JSTOP(4,1,K).NE.1.OR.JSTOP(4,2,K).NE.0) GO TO 21            UPDATE
      CALL IMPULS(4,4,K)                                              UPDATE
      I = -1                                                          UPDATE
   21 JSTOP(4,2,K) = JSTOP(4,1,K)                                     UPDATE
C                                                                     UPDATE
C     TEST TO LOCK OR UNLOCK JOINTS                                   UPDATE
C                                                                     UPDATE
C                                                                     UPDATE
C     CONDITIONS TO CHANGE SIGN OF IPIN(J)                            UPDATE
C                                                                     UPDATE
C                  PINNED               UNPINNED                      UPDATE
C         LOCKED (-1) !H.TQ! > T1    (-2) !TQ! > T1                   UPDATE
C                                                                     UPDATE
C      UNLOCKED  (+1) !H.TQ! < T2    (+2) !TQ! < T2                   UPDATE
C                          OR                 OR                      UPDATE
C                       WJ  < T3           WJ  < T3                   UPDATE
C                                                                     UPDATE
      DO 28 J=1,NJNT                                                  UPDATE
      IF (IABS(IPIN(J)).EQ.4) GO TO 28                                UPDATE
      IF (IPIN(J)) 22,28,23                                           UPDATE
   22 T1 = VISC(4,3*J-2)                                              UPDATE
      IF (T1.EQ.0.0) GO TO 28                                         UPDATE
      IF (IPIN(J).GT.-1) GOTO 51                                      SLIP
      IF (IPIN(J).GT.-6.AND.IPIN(J).LT.-1) GOTO 51                    SLIP
      TQM = XDY(HB(1,2*J),D(1,1,J+1),TQ(1,J))                         UPDATE
      ABSTQM = DABS(TQM)                                              UPDATE
      IF  (ABSTQM.GT.T1)  HA(2,2*J-1) = TQM                           UPDATE
      TQM = ABSTQM                                                    UPDATE
      GO TO 52                                                        UPDATE
   51 TQM = DSQRT(TQ(1,J)**2 + TQ(2,J)**2 + TQ(3,J)**2)               UPDATE
      IF  (TQM.GT.T1)  CALL DOT31(HIR(1,1,J),TQ(1,J),HA(1,2*J-1))     UPDATE
   52 IF  (TQM-T1)  28,28,26                                          UPDATE
   23 T2 = VISC(5,3*J-2)                                              UPDATE
      IF  (HA(2,2*J).NE.0.0)  GO TO 54                                UPDATE
      DO  53  K=1,3                                                   UPDATE
   53 HA(K,2*J-1) = 0.0                                               UPDATE
   54 IF  (T2.EQ.0.0)  GO TO 24                                       UPDATE
      IF (IPIN(J).GE.2.AND.IPIN(J).LE.5)                              SLIP
     *     TQM = DSQRT(TQ(1,J)**2+TQ(2,J)**2+TQ(3,J)**2)              SLIP
      IF (IPIN(J).EQ.1.OR.IPIN(J).EQ.6.OR.IPIN(J).EQ.7)               SLIP
     *     TQM = DABS(XDY(HB(1,2*J),D(1,1,J+1),TQ(1,J)))              SLIP
      IF (TQM-T2) 25,28,28                                            UPDATE
   24 T3 = VISC(6,3*J-2)                                              UPDATE
      IF (T3.EQ.0.0) GO TO 28                                         UPDATE
      IF (WJ(J)-T3) 25,28,28                                          UPDATE
   25 CALL IMPLS2(0,J,HB(1,2*J))                                      UPDATE
      I = -1                                                          UPDATE
   26 IPIN(J) = -IPIN(J)                                              UPDATE
      TMSEC = 1000.0*TIME                                             UPDATE
```

```
      IPINJ = -IPIN(J)                                                  UPDATE
      WRITE (6,27) TMSEC,J,IPINJ,IPIN(J)                                UPDATE
   27 FORMAT('0 AT TIME =',F9.3,' MSEC,    IPIN(',I2,                   BUTLER1
     *        ') HAS BEEN CHANGED FROM',I3,' TO',I3)                    BUTLER1
   28 CONTINUE                                                          UPDATE
C                                                                       UPDATE
C     TEST TO LOCK OR UNLOCK EULER JOINTS AXES.                         UPDATE
C     USE SAME TEST AS ABOVE BUT ON EACH AXIS SERARATELY.               UPDATE
C                                                                       UPDATE
C      IF LOCK(IEULER,K) IS NEGATIVE, AXIS K IS LOCKED;                 UPDATE
C         TO UNLOCK AXIS SET IEULER TO -LOCK(IEULER,K).                 UPDATE
C                                                                       UPDATE
C     IF LOCK(IEULER,K) IS POSITIVE, AXIS K IS UNLOCKED;                UPDATE
C       TO LOCK AXIS SET IEULER TO LOCK(IEULER,K).                      UPDATE
C                                                                       UPDATE
      DO 36 J=1,NJNT                                                    UPDATE
      IF (IABS(IPIN(J)).NE.4) GO TO 36                                  UPDATE
      JEULER = IEULER(J)                                                UPDATE
      CALL DOT31(HIR(1,1,J),TQ(1,J),TQTEST)                             UPDATE
      DO 31 K=1,3                                                       UPDATE
      K3J = 3*J-3+K                                                     UPDATE
      NLOCK = LOCK(JEULER,K)                                            UPDATE
      IF (NLOCK.GT.0) GO TO 29                                          UPDATE
      IF (VISC(4,K3J).EQ.0.0) GO TO 31                                  UPDATE
      IF (DABS(TQTEST(K)).LE.VISC(4,K3J)) GO TO 31                      UPDATE
      JEULER = -NLOCK                                                   UPDATE
      HA(K,2*J-1) = TQTEST(K)                                           UPDATE
      GO TO 31                                                          UPDATE
   29 IF  (HA(K,2*J).EQ.0.0)  HA(K,2*J-1) = 0.0                         UPDATE
      IF (VISC(5,K3J).EQ.0.0) GO TO 30                                  UPDATE
      IF (DABS(TQTEST(K)).LT.VISC(5,K3J)) JEULER =  NLOCK               UPDATE
      GO TO 31                                                          UPDATE
   30 IF (VISC(6,K3J).EQ.0.0) GO TO 31                                  UPDATE
      IF (DABS(ANGD(K,J)).LT.VISC(6,K3J)) JEULER =  NLOCK               UPDATE
   31 CONTINUE                                                          UPDATE
      IF (JEULER.EQ.IEULER(J)) GO TO 36                                 UPDATE
      TMSEC = 1000.0*TIME                                               UPDATE
      WRITE (6,32) TMSEC,J,IEULER(J),JEULER                             UPDATE
   32 FORMAT('0 AT TIME =',F9.3,' MSEC, IEULER(',I2,                    BUTLER1
     *        ') HAS BEEN CHANGED FROM',I3,' TO',I3)                    BUTLER1
      IF (JEULER.EQ.8)          GO TO 35                                UPDATE
      IF (IEULER(J).EQ.7) GO TO 35                                      UPDATE
      IF (IEULER(J).EQ.6 .AND. (JEULER.EQ.2.OR.JEULER.EQ.1)) GO TO 35   UPDATE
      IF (IEULER(J).EQ.5 .AND. (JEULER.EQ.3.OR.JEULER.EQ.1)) GO TO 35   UPDATE
      IF (IEULER(J).EQ.4 .AND. (JEULER.EQ.3.OR.JEULER.EQ.2)) GO TO 35   UPDATE
      MODE = -1                                                         UPDATE
      K = JEULER                                                        UPDATE
      IF (K.GT.3) GO TO 33                                              UPDATE
      IF (K.EQ.2) GO TO 34                                              UPDATE
      K4 = 4-K                                                          UPDATE
```

```
      CALL CROSS (HIR(1,K4,J),HIR(1,2,J),T)                             UPDATE
      IEULER(J) = 8                                                     UPDATE
      IPIN(J)   = 4                                                     UPDATE
      CALL IMPLS2(MODE,J,T)                                             UPDATE
      I = -1                                                            UPDATE
      GO TO 35                                                          UPDATE
   33 MODE =  1                                                         UPDATE
      K = K-3                                                           UPDATE
      IF (K.GT.3) MODE=0                                                UPDATE
   34 IEULER(J) = 8                                                     UPDATE
      IPIN(J)   = 4                                                     UPDATE
      CALL IMPLS2(MODE,J,HIR(1,K,J))                                    UPDATE
      I =  1                                                            UPDATE
   35 IEULER(J) = JEULER                                                UPDATE
      IPIN(J) = 4                                                       UPDATE
      IF (IEULER(J).NE.8) IPIN(J) = -4                                  UPDATE
C     GET SINE AND COSINE OF NUTATION IF IEULER GOES TO STATE 2         JDRIFT
      CALL EJOINT(-1,J)                                                 JDRIFT
      IF(JEULER.NE.2) GOTO 36                                           JDRIFT
      TQM=ANG(2,J)+CONST(2,J)                                           JDRIFT
      CONST(4,J) = DCOS(TQM)                                            JDRIFT
      CONST(5,J) = DSIN(TQM)                                            JDRIFT
   36 CONTINUE                                                          UPDATE
      DO 90 J = 1,NJNT                                                  SLIP
      IF (IABS(IPIN(J)).LE.4) GO TO 90                                  SLIP
      IF (IEULER(J).GE.0) GO TO 90                                      SLIP
      IF (CONST(1,J).EQ.0.0.AND.CONST(2,J).EQ.0.0) GO TO 90             SLIP
      M = JNT(J)                                                        SLIP
      FTEST = XDY(HT(1,3,2*J-1),D(1,1,M),F(1,J))                        SLIP
      IF (FTEST.GE.CONST(1,J).AND.FTEST.LE.CONST(2,J)) GO TO 90         SLIP
      IEULER(J) = 0                                                     SLIP
      TMSEC = 1000.0*TIME                                               SLIP
      WRITE (6,88) TMSEC,J                                              SLIP
   88 FORMAT(/'0 AT TIME =',F9.3,' MSEC,  JOINT ',I3,' HAS BEEN',       SLIP
     *  ' UNLOCKED AND ALLOWED TO SLIP.'/)                              SLIP
   90 CONTINUE                                                          SLIP
C     F IS THE FORCE ON SEGMENT J+1, - F IS ON SEGMENT M                SLIP
C                                                                       UPDATE
   37 IF  (NQ.LE.0)  GO TO 41                                           UPDATE
      DO 40 K=1,NQ                                                      UPDATE
      IF (KQTYPE(K).LT.3) GO TO 40                                      UPDATE
      IF (KQTYPE(K).GT.4) GO TO 40                                      UPDATE
      IF (CFQQ(K).LT.0.0) KQTYPE(K) = -KQTYPE(K)                        UPDATE
      IF (CFQQ(K).LT.0.0) GO TO 39                                      UPDATE
C                                                                       UPDATE
C     TEST IF ROLLING CONSTRAINT SHOULD BE SLIDING AND VICE VERSA.      UPDATE
C                                                                       UPDATE
      QN = -XDY(TQQ(1,K),HHT(1,1,K),QQ(1,K))                            UPDATE
      IF (NPRT(24).NE.0) WRITE (6,38) KQTYPE(K),KQ1(K),KQ2(K),          UPDATE
     *                  (RK1(II,K),II=1,3),(RK2(II,K),II=1,3),          UPDATE
```

```
     *                  ((HHT(II,J,K),J=1,3),II=1,3),                     UPDATE
     *        (QQ(II,K),II=1,3),(TQQ(II,K),II=1,3),(RQQ(II,K),II=1,3),    UPDATE
     *                (HQQ(II,K),II=1,3),SQQ(K),CFQQ(K),QN                UPDATE
  38 FORMAT('0 UPDATE ROLL-SLIDE TEST'/(2X,9G14.6))                       UPDATE
     IF (QN.LT.0.0) KQTYPE(K) = -4                                        UPDATE
     IF (QN.LT.0.0) GO TO 39                                              UPDATE
     QDOTQ = QQ(1,K)**2 + QQ(2,K)**2 + QQ(3,K)**2                         UPDATE
     QT = DSQRT(QDOTQ-QN**2)                                              UPDATE
     IF (KQTYPE(K).EQ.3 .AND. QT.LE.CFQQ(K)*QN) GO TO 40                  UPDATE
     IF (KQTYPE(K).EQ.4 .AND. QT.GE.0.9*CFQQ(K)*QN) GO TO 40              UPDATE
     KQTYPE(K) = 7-KQTYPE(K)                                              UPDATE
  39 CALL OUTPUT(0)                                                       UPDATE
     CALL SETUP2                                                          UPDATE
     CALL DAUX(K)                                                         UPDATE
     IF  (NPRT(24).NE.0)  CALL OUTPUT(1)                                  UPDATE
     IF  (NPRT( 3).NE.0)  CALL PRINT (6HUPDATE)                           UPDATE
     I = -1                                                               UPDATE
  40 CONTINUE                                                             UPDATE
  41 CALL ELTIME(2,7)                                                     UPDATE
  42 RETURN                                                               UPDATE
     END                                                                  UPDATE
```

```
      SUBROUTINE  UPDFDC  (M)                                           UPDFDC
C                                               REV III.2 08/08/84REVIII
C     UPDATE FORCE DEFLECTION CURVE DEFINITION THAT IS DEFINED          UPDFDC
C     IN LOCATION M OF NTAB ARRAY.  SUBROUTINE ASSUMES THAT             UPDFDC
C     A SUCCESSFUL INTEGRATION STEP HAS JUST BEEN COMPLETED AND         UPDFDC
C     WILL COMPUTE ENTIRE CURVE DEFINITION TO BE VALID FOR NEXT STEP.   UPDFDC
C                                                                       UPDFDC
      IMPLICIT REAL*8(A-H,O-Z)                                          UPDFDC
      COMMON/TABLES/MXNTI,MXNTB,MXTB1,MXTB2,NTI(50),NTAB(1250),TAB(4500)DIMENB
       L      = NTAB(M)                                                 UPDFDC
      IF (L.EQ.0)  GO TO 99                                             UPDFDC
      D       = TAB(L)                                                  UPDFDC
      IF (D.LT.0.0) D = 0.0                                             UPDFDC
      DLAST  = TAB(L+1)                                                 UPDFDC
      IF (D.EQ.DLAST)  GO TO 99                                         UPDFDC
      DCUBIC = TAB(L+6)                                                 UPDFDC
      IF (D.EQ.DCUBIC) GO TO 98                                         UPDFDC
      AREA   = TAB(L+2)                                                 UPDFDC
      RLAST  = TAB(L+3)                                                 UPDFDC
      GLAST  = TAB(L+4)                                                 UPDFDC
      DG     = TAB(L+5)                                                 UPDFDC
      DG0    = DG                                                       UPDFDC
      DREF   = TAB(L+7)                                                 UPDFDC
      DMAX   = TAB(L+8)                                                 UPDFDC
      DINER  = TAB(L+9)                                                 UPDFDC
      FDMAX  = TAB(L+10)                                                UPDFDC
      DC0    = TAB(L+18)                                                UPDFDC
      LQ     = L+11                                                     UPDFDC
      LC     = L+14                                                     UPDFDC
      IF (NTAB(M+1).LT.0) GO TO 98                                      UPDFDC
      IF (D-DCUBIC) 10,98,20                                            UPDFDC
C                                                                       UPDFDC
C     D < DCUBIC, DEFINE NEW CUBIC                                      UPDFDC
C     Y(X) = A0 + A1*(X-X1) + A2*(X-X1)**2 + A3*(X-X1)**3               UPDFDC
C     WHOSE DERIVATIVE IS                                               UPDFDC
C     Y'(X) = A1 + 2*A2*(X-X1) + 3*A3+(X-X1)**2                         UPDFDC
C                                                                       UPDFDC
   10 X1 = DMAX1 (D,DG)                                                 UPDFDC
      X2 = DREF                                                         UPDFDC
C                                                                       UPDFDC
C     IF INERTIAL SPIKE EXISTS AND IF DIMAX < DREF , DROP INERTIAL SPIKEUPDFDC
      NI = NTAB(M+2)                                                    UPDFDC
      IF (NI.GT.0.AND.TAB(NI+3).GT.0.0.AND.DREF.GT.TAB(NI+3))NTAB(M+2)=0UPDFDC
      DX = X2-X1                                                        UPDFDC
      X = X1-DG                                                         UPDFDC
      Y1 = TAB(LQ) +X *(TAB(LQ+1)+X *TAB(LQ+2))                         UPDFDC
      Y1P = TAB(LQ+1)+2.0*X *TAB(LQ+2)                                  UPDFDC
      X2DOT = 0.0                                                       UPDFDC
      CALL FRCDFL (X2,X2DOT,M,0,Y2P,ELOSS)                              UPDFDC
      CALL FRCDFL (X2,X2DOT,M,1,Y2 ,ELOSS)                              UPDFDC
```

```
      DCUBIC = X1                                                       UPDFDC
      DC0    = DCUBIC                                                   UPDFDC
C                                                                       UPDFDC
C     A0 = Y(X1)   (THE VALUE OF THE QUADRATIC AT X1)                   UPDFDC
C     A1 = Y'(X1)  (THE DERIVATIVE OF THE QUADRATIC AT X1)              UPDFDC
C                                                                       UPDFDC
      A0 = Y1                                                           UPDFDC
      A1 = Y1P                                                          UPDFDC
C                                                                       UPDFDC
C     SOLVE SIMULTANEOUSLY FOR A2 AND A3                                UPDFDC
C        A2*(X2-X1)**2 +    A3*(X2-X1)**3 = Y(X2)-A0-A1*(X2-X1)         UPDFDC
C      2*A2*(X2-X1)     +  3*A3*(X2-X1)**2 = Y'(X2)-A1                  UPDFDC
C                                                                       UPDFDC
      R13 = (Y2 - Y1 -Y1P*DX)/DX**2                                     UPDFDC
      R23 = (Y2P - Y1P)/DX                                              UPDFDC
      A2  = 3.0*R13 - R23                                               UPDFDC
      A3  = (R23 - 2.0*R13)/DX                                          UPDFDC
C                                                                       UPDFDC
C     IF LOCAL MINIMUN OF CUBIC (ABSCISSA VALUE WHERE Y'(X) = 0)        UPDFDC
C     LIES BETWEEN DCUBIC AND DREF AND IS NEGATIVE, THEN REPLACE        UPDFDC
C     CUBIC DEFINITION WITH STRAIGHT LINE BETWEEN (X1,Y1) AND (X2,Y2).  UPDFDC
C                                                                       UPDFDC
      IF (A3.NE.0.0)  GO TO 14                                          UPDFDC
      R2 = -0.5*A1/A2                                                   UPDFDC
      GO TO 15                                                          UPDFDC
   14 A33 = 3.0*A3                                                      UPDFDC
      DISC = A2**2-A1*A33                                               UPDFDC
      IF (DISC.LT.0.0)  GO TO 13                                        UPDFDC
      SQDISC = DSQRT(DISC)                                              UPDFDC
      R1 = (-A2+SQDISC)/A33                                             UPDFDC
      IF (R1.LE.0.0.OR.R1.GE.DX)  GO TO 11                              UPDFDC
      FR1 = A0+R1*(A1+R1*(A2+R1*A3))                                    UPDFDC
      IF (FR1.LT.0.0) GO TO 12                                          UPDFDC
   11 R2 = (-A2-SQDISC)/A33                                             UPDFDC
   15 IF (R2.LE.0.0.OR.R2.GE.DX) GO TO 13                               UPDFDC
      FR2 = A0+R2*(A1+R2*(A2+R2*A3))                                    UPDFDC
      IF (FR2.GE.0.0)  GO TO 13                                         UPDFDC
   12 A0 = Y1                                                           UPDFDC
      A1 = (Y2-Y1)/DX                                                   UPDFDC
      A2 = 0.0                                                          UPDFDC
      A3 = 0.0                                                          UPDFDC
   13 TAB(LC) = A0                                                      UPDFDC
      TAB(LC+1) = A1                                                    UPDFDC
      TAB(LC+2) = A2                                                    UPDFDC
      TAB(LC+3) = A3                                                    UPDFDC
      TAB(L +6) = DCUBIC                                                UPDFDC
      TAB(L+18) = DC0                                                   UPDFDC
      GO TO 98                                                          UPDFDC
   20 IF (D-DREF) 21,21,30                                              UPDFDC
C                                                                       UPDFDC
```

```
C     DCUBIC < D < DREF, DEFINE NEW QUADRATIC FROM CUBIC CURVE.          UPDFDC
C                                                                        UPDFDC
   21 X = D-DCO                                                          UPDFDC
      Y2 = TAB(LC)+X*(TAB(LC+1)+X*(TAB(LC+2)+X*TAB(LC+3)))               UPDFDC
      X1 = DCUBIC - DG                                                   UPDFDC
      AREA = X1*(TAB(LQ)+X1*(TAB(LQ+1)/2.0+X1*TAB(LQ+2)/3.0))            UPDFDC
     * + X*(TAB(LC)+X*(TAB(LC+1)/2.0+X*(TAB(LC+2)/3.0+X*TAB(LC+3)/4.0)))UPDFDC
      X = DCUBIC - DCO                                                   UPDFDC
      IF (X.NE.0.0) AREA = AREA                                          UPDFDC
     * - X*(TAB(LC)+X*(TAB(LC+1)/2.0+X*(TAB(LC+2)/3.0+X*TAB(LC+3)/4.0)))UPDFDC
      GO TO 31                                                           UPDFDC
C                                                                        UPDFDC
C     DREF < D, DEFINE NEW QUADRATIC FROM BASE CURVE.                    UPDFDC
C                                                                        UPDFDC
C     IF DINER < D , REMOVE INERTIAL SPIKE                               UPDFDC
C                                                                        UPDFDC
   30 IF  (NTAB(M+2).GT.0 .AND. D.GE.DINER)  NTAB(M+2) = 0               UPDFDC
      NR = NTAB(M+3)                                                     UPDFDC
      RLAST = 1.0                                                        UPDFDC
      IF (NR.GT.0  ) RLAST = EVALFD(D,NR,1)                              UPDFDC
      IF (RLAST.NE.1.0) GO TO 39                                         UPDFDC
C                                                                        UPDFDC
C     R = 1,  USE BASE CURVE FOR UNLOADING                               UPDFDC
C                                                                        UPDFDC
      DG = 0.0                                                           UPDFDC
      DCUBIC = 0.0                                                       UPDFDC
      DREF = 0.0                                                         UPDFDC
      A0 = 0.0                                                           UPDFDC
      A1 = 0.0                                                           UPDFDC
      A2 = 0.0                                                           UPDFDC
      GO TO 32                                                           UPDFDC
   39 NG = NTAB(M+4)                                                     UPDFDC
      GLAST = 0.0                                                        UPDFDC
      IF (NG.GT.0  ) GLAST = EVALFD(D,NG,1)                              UPDFDC
      NB =  NTAB(M+1)                                                    UPDFDC
      D0 = TAB(NB)                                                       UPDFDC
      DG = D0 + GLAST*(D-D0)                                             UPDFDC
      Y2 = EVALFD(D, NB,1)                                               UPDFDC
      NI = NTAB(M+2)                                                     UPDFDC
      IF (NI.GT.0) Y2 = Y2+EVALFD(D,NI,1)                                UPDFDC
      AREA = EVALFD(D,NB,2)                                              UPDFDC
      DREF = D                                                           UPDFDC
   31 DCUBIC = D                                                         UPDFDC
      X1 = DG                                                            UPDFDC
      X2 = D                                                             UPDFDC
      DX = X2-X1                                                         UPDFDC
      Y1 = 0.0                                                           UPDFDC
      RAREA = RLAST*AREA                                                 UPDFDC
C                                                                        UPDFDC
C     COMPUTE UNLOADING QUADRATIC COEFFICIENTS SUCH THAT                 UPDFDC
```

```
C     ENDPOINT DERIVATES ARE NON-NEGATIVE.                                     UPDFDC
C                                                                              UPDFDC
      A0   = 0.0                                                               UPDFDC
      A1   = 2.0/DX*(3.0*RAREA/DX-Y2)                                          UPDFDC
      IF (A1.LT.0.0)  A1 = 0.0                                                 UPDFDC
      A2 = (Y2/DX-A1)/DX                                                       UPDFDC
      IF (A2.GE.0.0) GO TO 32                                                  UPDFDC
      A1 = Y2/DX                                                               UPDFDC
      A2 = 0.0                                                                 UPDFDC
C                                                                              UPDFDC
C     RESTORE TAB VALUES THAT MAY HAVE BEEN CHANGED                            UPDFDC
C                                                                              UPDFDC
   32 TAB(L+2) = AREA                                                          UPDFDC
      TAB(L+3) = RLAST                                                         UPDFDC
      TAB(L+4) = GLAST                                                         UPDFDC
      TAB(L+5) = DG                                                            UPDFDC
      TAB(L+6) = DCUBIC                                                        UPDFDC
      TAB(L+7) = DREF                                                          UPDFDC
      TAB(LQ)    = A0                                                          UPDFDC
      TAB(LQ+1) = A1                                                           UPDFDC
      TAB(LQ+2) = A2                                                           UPDFDC
   98 TAB(L+1) = D                                                             UPDFDC
      IF (D.GT.DG0 .AND. DLAST.LE.DG0) M=-M                                    UPDFDC
   99 RETURN                                                                   UPDFDC
      END                                                                      UPDFDC
```

```
      SUBROUTINE VEHPOS                                                 VEHPOS
C                                                  REV IV    07/23/86TWOPI
C     COMPUTES COMPONENTS OF VEHICLE ACCELERATIONS ONLY AS A FUNCTION   VEHPOS
C     OF TIME USING DATA AND TABLES PRODUCED BY SUBROUTINE VINPUT.      VEHPOS
C                                                                       VEHPOS
      IMPLICIT REAL*8 (A-H,O-Z)                                         VEHPOS
      COMMON/CONTRL/ TIME,NSEG,NJNT,NPL,NBLT,NBAG,NVEH,NGRND,           VEHPOS
     *               NS,NQ,NSD,NFLX,NHRNSS,NWINDF,NJNTF,NPRT(36),NPG    PAGE
      COMMON/SGMNTS/ D(3,3,30),WMEG(3,30),WMEGD(3,30),U1(3,30),U2(3,30),VEHPOS
     *               SEGLP(3,30),SEGLV(3,30),SEGLA(3,30),NSYM(30)       VEHPOS
      COMMON/VPOSTN/ ZPLT(3),SPLT(3),AXV(3,6),VATAB(6,501,6),           VEHICL
     *               VT0(6),VDT(6),TIMEV(6),OMEGV(6),NVTAB(6),INDXV(6)  VEHPOS
      COMMON/CNSNTS/ PI,RADIAN,G,THIRD,EPS(24),                         VEHPOS
     *               UNITL,UNITM,UNITT,GRAVTY(3),TWOPI                  TWOPI
      DIMENSION AX(3)                                                   VEHPOS
      T = TIME                                                          VEHPOS
      M = 1                                                             VEHPOS
   15 DO  16  I=1,3                                                     VEHPOS
   16 AX(I) = AXV(I,M)                                                  VEHPOS
      AT0 = VT0(M)                                                      VEHPOS
      ADT = VDT(M)                                                      VEHPOS
      VTIME = TIMEV(M)                                                  VEHPOS
      OMEG  = OMEGV(M)                                                  VEHPOS
      NATAB = NVTAB(M)                                                  VEHPOS
      K = INDXV(M)                                                      VEHPOS
      IF(NATAB.NE.0) GO TO 20                                           VEHPOS
C                                                                       VEHPOS
C     HALF-SINE WAVE DECELERATION                                       VEHPOS
C                                                                       VEHPOS
      IF(T.GT.VTIME) T=VTIME                                            VEHPOS
      WT = OMEG*T                                                       VEHPOS
      SWT = DSIN(WT)                                                    VEHPOS
      DO 10 I=1,3                                                       VEHPOS
      AW = AX(I)*OMEG                                                   VEHPOS
      SEGLA(I,K) = -AW*OMEG*SWT                                         VEHPOS
   10 WMEGD(I,K) = 0.0                                                  VEHPOS
      GO TO 99                                                          VEHPOS
   20 IF (NATAB.LT.0) GO TO 30                                          VEHPOS
C                                                                       VEHPOS
C      UNIDIRECTIONAL DECELERATION                                      VEHPOS
C                                                                       VEHPOS
      IF (T.LT.VTIME) GO TO 21                                          VEHPOS
C                                                                       VEHPOS
C     TIME POINT EXCEEDS TABLE, USE LAST VALUES OF ACCELERATION.        VEHPOS
C                                                                       VEHPOS
      AC0 = VATAB(1,NATAB,M)                                            VEHPOS
      GO TO 25                                                          VEHPOS
C                                                                       VEHPOS
C         USE QUADRATIC INTERPOLATION FROM TABLES FOR CURRENT VALUE OF  VEHPOS
C         TIME TO BE CONSISTENT WITH SIMPSON INTEGRATION OF TABLES.     VEHPOS
```

```
C                                                                         VEHPOS
   21 J= 0.5*(T-ATO)/ADT +1.0                                             VEHPOS
      XK = T/ADT -DFLOAT(2*J-1)                                           VEHPOS
      X1 = XK+1.0                                                         VEHPOS
      X3 = XK-1.0                                                         VEHPOS
      ACO = 0.5*XK*X3*VATAB(1,2*J-1,M)                                    VEHPOS
     *    -       X3*X1*VATAB(1,2*J  ,M)                                  VEHPOS
     *      + 0.5*XK*X1*VATAB(1,2*J+1,M)                                  VEHPOS
C                                                                         VEHPOS
C         COMPONENTS OF VEHICLE ACCELERATION.                             VEHPOS
C                                                                         VEHPOS
   25 DO 29 I=1,3                                                         VEHPOS
      SEGLA(I,K) = -G*AX(I)*ACO                                           VEHPOS
   29 WMEGD(I,K) = 0.0                                                    VEHPOS
      GO TO 99                                                            VEHPOS
C                                                                         VEHPOS
C       OMNIDIRECTIONAL DECELERATION                                      VEHPOS
C                                                                         VEHPOS
   30 J = (TIME-ATO)/ADT + 1.0                                            VEHPOS
      IF (J.GE.-NATAB) GO TO 32                                           VEHPOS
C                                                                         VEHPOS
C          INTERPOLATION FROM VINPUT TABLES OF COMPONENTS OF VEHICLE      VEHPOS
C          LINEAR AND ANGULAR ACCELERATION.                               VEHPOS
C                                                                         VEHPOS
      TJ        = ATO + DFLOAT(J-1)*ADT                                   VEHPOS
      DLT       = TIME-TJ                                                 VEHPOS
      R1 = DLT/ADT                                                        VEHPOS
      R2 = 1.0-R1                                                         VEHPOS
      DO 31 I=1,3                                                         VEHPOS
      SEGLA(I,K) =      -G*(VATAB(I  ,J+1,M)*R1 + VATAB(I  ,J,M)*R2)      VEHPOS
   31 WMEGD(I,K) = RADIAN*(VATAB(I+3,J+1,M)*R1 + VATAB(I+3,J,M)*R2)       VEHPOS
      GO TO 99                                                            VEHPOS
C                                                                         VEHPOS
C     TIME POINT EXCEEDS TABLE, USE LAST VALUES OF ACCELERATION.          VEHPOS
C                                                                         VEHPOS
   32 J = - NATAB                                                         VEHPOS
      DO 33 I=1,3                                                         VEHPOS
      SEGLA(I,K) =      -G*VATAB(I  ,J,M)                                 VEHPOS
   33 WMEGD(I,K) = RADIAN*VATAB(I+3,J,M)                                  VEHPOS
   99 M = M+1                                                             VEHPOS
      IF  (M.LE.6 .AND. INDXV(M).NE.0)  GO TO 15                          VEHPOS
      RETURN                                                              VEHPOS
      END                                                                 VEHPOS
```

```
      SUBROUTINE VINPUT                                                 VINPUT
C                                                   REV IV    07/24/86SLIP
C     PERFORMS CARD INPUT AND COMPUTES DATA AND TABLES REQUIRED BY      VINPUT
C     SUBROUTINE VEHPOS TO INTEGRATE THE CRASH VEHICLE MOTION FOR ONE OFVINPUT
C     THREE PERMISSABLE OPTIONS:                                        VINPUT
C       (1) HALF SINE-WAVE LINEAR DECELERATION IMPULSE                  VINPUT
C       (2) UNIDIRECTIONAL LINEAR DECELERATION TABULAR INPUT            VINPUT
C       (3) OMNIDIRECTIONAL LINEAR AND ANGULAR ACCELERATION TABULAR     VINPUT
C           INPUT (6 DEGREES OF FREEDOM VEHICLE MOTION)                 VINPUT
C                                                                       VINPUT
      IMPLICIT REAL*8 (A-H,O-Z)                                         VINPUT
      COMMON/CONTRL/ TIME,NSEG,NJNT,NPL,NBLT,NBAG,NVEH,NGRND,           VINPUT
     *               NS,NQ,NSD,NFLX,NHRNSS,NWINDF,NJNTF,NPRT(36),NPG    PAGE
      COMMON/SGMNTS/ D(3,3,30),WMEG(3,30),WMEGD(3,30),U1(3,30),U2(3,30),VINPUT
     *               SEGLP(3,30),SEGLV(3,30),SEGLA(3,30),NSYM(30)       VINPUT
      COMMON/DESCRP/ PHI(3,30),W(30),RW(30),SR(4,60),HA(3,60),HB(3,60), SLIP
     *               RPHI(3,30),HT(3,3,60),SPRING(5,90),VISC(7,90),     VINPUT
     *               JNT(30),IPIN(30),ISING(30),IGLOB(30),JOINTF(30)    VINPUT
      COMMON/VPOSTN/ ZPLT(3),SPLT(3),AXV(3,6),VATAB(6,501,6),           VEHICL
     *               VT0(6),VDT(6),TIMEV(6),OMEGV(6),NVTAB(6),INDXV(6)  VINPUT
      COMMON/TEMPVS/ X0(3),XDOT0(3),XCOMP(3),XVCOMP(3),ANGLE(3),        VINPUT
     *               ATAB(15,501),DVEH(3,3),VMEG(3),VMEGD(3),           VEHICL
     *              XACOMP(3),THET(3),AX(3),F(5,101),XYZ(103,6),TT(103),CHGIII
     *               VIPS,VMPH,AT0,ADT,VTIME,OMEG,NATAB                 VINPUT
     *              ,SP(5,101,4),Q1(101,4),A1(3),W1(4),QD(4),QC(4)      JTF984
      COMMON/INTEST/ SGTEST(3,4,30),XTEST(3,120),SEGT(120),REGT(120)    VINPUT
      REAL    SEGT                                                      VINPUT
      COMMON/CNSNTS/ PI,RADIAN,G,THIRD,EPS(24),                         VINPUT
     *               UNITL,UNITM,UNITT,GRAVTY(3),TWOPI                  TWOPI
      COMMON/TITLES/ DATE(3),COMENT(40),VPSTTL(20),BDYTTL(5),           VINPUT
     *               BLTTTL(5,8),PLTTL(5,30),BAGTTL(5,6),SEG(30),       VINPUT
     *               JOINT(30),CGS(30),JS(30)                           VINPUT
      REAL DATE,COMENT,VPSTTL,BDYTTL,BLTTTL,PLTTL,BAGTTL,SEG,JOINT      VINPUT
      LOGICAL*1 CGS,JS                                                  VINPUT
      DIMENSION IDYPR(3)                                                VINPUT
      REAL VEH(6),GRND                                                  VINPUT
      DATA VEH/4HVEH1,4HVEH2,4HVEH3,4HVEH4,4HVEH5,4HVEH /,GRND/4HGRND/  VINPUT
      DATA IDYPR/3,2,1/                                                 VINPUT
      DATA MXTAB2/99/,MXTAB3/501/,MXTAB4/101/                           MISC
C                                                                       VINPUT
C     READ AND PRINT CONTENTS OF CARDS C.1 AND C.2                      VINPUT
C                                                                       VINPUT
      NVEH = NSEG                                                       VINPUT
      NVH  = 0                                                          VINPUT
      DO   11  I=1,6                                                    VINPUT
   11 INDXV(I) = 0                                                      VINPUT
   12 READ (5,13) VPSTTL                                                VINPUT
   13 FORMAT (20A4)                                                     VINPUT
      READ(5,14) ANGLE,VIPS,VTIME,X0,NATAB,AT0,ADT,MSEG                 VINPUT
   14 FORMAT(8F6.0,I6,2F6.0,I6)                                         VINPUT
```

```
      INTAB = IABS(NATAB)                                               CHGIII
      IF (NATAB.GT.0.AND.INTAB.GT.MXTAB2) STOP 79                       MISC
      WRITE (6,15) NPG,VPSTTL,ANGLE,VIPS,VTIME,X0,NATAB,AT0,ADT,MSEG    PAGE
      NPG = NPG+1                                                       PAGE
   15 FORMAT('1 VEHICLE DECELERATION INPUTS',94X,'PAGE',I5/120X,        PAGE
     *       'CARDS C'/3X,20A4//                                        PAGE
     *    7X,'YAW',9X,'PITCH',7X,'ROLL',8X,'VIPS',8X,'VTIME',7X,'X0(X)',VINPUT
     *    7X,'X0(Y)',7X,'X0(Z)',2X,'NATAB',6X,'AT0',9X,'ADT',4X,'MSEG'/ VINPUT
     *    8F12.3,I5,2X,2F12.6,I5)                                       VINPUT
      DA1 = ANGLE(1)*RADIAN                                             VINPUT
      DA2 = ANGLE(2)*RADIAN                                             VINPUT
      AX(3) = DCOS(DA2)                                                 VINPUT
      AX(1) = DCOS(DA1)*AX(3)                                           VINPUT
      AX(2) = DSIN(DA1)*AX(3)                                           VINPUT
      AX(3) = DSIN(DA2)                                                 VINPUT
      IF(NATAB.NE.0) GO TO 18                                           VINPUT
C                                                                       VINPUT
C     HALF-SINE WAVE DECELERATION                                       VINPUT
C                                                                       VINPUT
      OMEG = PI/VTIME                                                   VINPUT
      AT = 0.5*VIPS/OMEG                                                VINPUT
      IF (VIPS.LT.0.0) VIPS = 0.0                                       VINPUT
      DO 16 I=1,3                                                       VINPUT
      XACOMP(I) = 0.0                                                   VINPUT
      XDOT0(I) = VIPS*AX(I)                                             VINPUT
   16 AX(I) = AT*AX(I)                                                  VINPUT
      WRITE (6,17) VIPS,UNITL,UNITT,ANGLE,VTIME,UNITT                   VINPUT
   17 FORMAT('0 PASSENGER COMPARTMENT DISPLACEMENT HISTORY'/            VINPUT
     *       '  ANALYTICAL HALF-SINE WAVE DECELERATION'/                VINPUT
     *       '  V0=',F8.3,1X,A4,'/',A4,', OBLIQUE ANGLES =',3F7.2,      VINPUT
     *       ' DEGREES, TIME DURATION =',F7.3,1X,A4//)                  VINPUT
      GO TO 28                                                          VINPUT
   18 IF (NATAB.LT.0) GO TO 31                                          VINPUT
C                                                                       VINPUT
C     FOR UN    ECTIONAL VEHICLE MOTION                                 VINPUT
C     READ LI   AR DECELERATION TABLES FROM CARDS C.3                   VINPUT
C                                                                       VINPUT
      READ (5,19) (ATAB(1,I),I=1,NATAB)                                 VINPUT
   19 FORMAT (12F6.0)                                                   VINPUT
C                                                                       VINPUT
C     EXTEND TABLE IF NECESSARY SUCH THAT NATAB IS ODD AND              VINPUT
C     LAST ENTRY NEED NOT BE ZERO. IF TABLE SIZE IS EXCEEDED ON TIME,   VINPUT
C     VALUE OF LAST ENTRY WILL BE USED.                                 VINPUT
C                                                                       VINPUT
      IF (MOD(NATAB,2).EQ.1) GO TO 20                                   VINPUT
      ATAB(1,NATAB+1) = ATAB(1,NATAB)                                   VINPUT
      NATAB = NATAB+1                                                   VINPUT
   20 VTIME = ADT * DFLOAT(NATAB-1)                                     VINPUT
C                                                                       VINPUT
C     USING SIMPSON'S INTEGRATION, COMPUTE VELOCITY AND DISPLACEMENT    VINPUT
```

```
C     TABLE FOR NATAB EQUALLY SPACED (ADT) TIME POINTS.                     VINPUT
C     FOR I=1,NATAB                                                         VINPUT
C       ATAB(1,I) = LINEAR DECELERATION (G'S)                               VINPUT
C       ATAB(2,I) = LINEAR VELOCITY (L UNITS/T UNITS)                       VINPUT
C       ATAB(3,I) = LINEAR DISPLACEMENT (L UNITS)                           VINPUT
C                                                                           VINPUT
      ATAB(2,1) = VIPS                                                      VINPUT
      ATAB(3,1) = 0.0                                                       VINPUT
      DA1 = ADT/3.0                                                         VINPUT
      DA2 = ADT/12.0                                                        VINPUT
      UNITS = -G                                                            VINPUT
      DO 22 J=2,3                                                           VINPUT
      DO 21 I=2,NATAB,2                                                     VINPUT
      F1 = ATAB(J-1,I-1) * UNITS                                            VINPUT
      F2 = ATAB(J-1,I  ) * UNITS                                            VINPUT
      F3 = ATAB(J-1,I+1) * UNITS                                            VINPUT
      ATAB(J,I  ) = ATAB(J,I-1) + DA2*(5.0*F1+8.0*F2-F3)                    VINPUT
   21 ATAB(J,I+1) = ATAB(J,I-1) + DA1*(    F1+4.0*F2+F3)                    VINPUT
   22 UNITS = 1.0                                                           VINPUT
C                                                                           VINPUT
C     PRINT TABLES                                                          VINPUT
C                                                                           VINPUT
      WRITE (6,23) (UNITL,UNITT,UNITL,I=1,2)                                VINPUT
   23 FORMAT('0 UNIDIRECTIONAL VEHICLE POSITION TABLES'//                   VINPUT
     *     2('     TIME       ACC       VELOCITY      POSITION    ')/        VINPUT
     *   2('     (MSEC)       (G)     (',A4,'/',A4,')',5X,'(',A4,')',4X)/)VINPUT
      DO 26 J=1,50                                                          VINPUT
      IF (J.GT.NATAB) GO TO 26                                              VINPUT
      T1 = (ATO + DFLOAT(J-1)*ADT)*1000.0                                   VINPUT
      IF (J+50.LE.NATAB) GO TO 25                                           VINPUT
      WRITE (6,24) T1,(ATAB(I,J),I=1,3)                                     VINPUT
   24 FORMAT(2(F11.5,F10.2,F13.4,F13.5,3X))                                 VINPUT
      GO TO 26                                                              VINPUT
   25 T2 = (ATO + DFLOAT(J+49)*ADT)*1000.0                                  VINPUT
      WRITE (6,24) T1,(ATAB(I,J),I=1,3),T2,(ATAB(I,J+50),I=1,3)             VINPUT
   26 CONTINUE                                                              VINPUT
C                                                                           VINPUT
C     INITIALIZATION                                                        VINPUT
C                                                                           VINPUT
      DO 27 I=1,3                                                           VINPUT
      XACOMP(I) = -G*AX(I)*ATAB(1,1)                                        VINPUT
   27 XDOTO(I)= VIPS*AX(I)                                                  VINPUT
   28 DO 30 I=1,3                                                           VINPUT
      DO 29 J=1,3                                                           VINPUT
   29 DVEH(I,J) = 0.0                                                       VINPUT
      DVEH(I,I) = 1.0                                                       VINPUT
      VMEGD(I) = 0.0                                                        VINPUT
   30 VMEG(I)   = 0.0                                                       VINPUT
      GO TO 64                                                              VINPUT
C                                                                           VINPUT
```

```
C     FOR OMNIDIRECTIONAL (6 DEGREES OF FREEDOM) VEHICLE MOTION          VINPUT
C     READ LINEAR DECELERATION AND ANGULAR ACCELERATION TABLES           VINPUT
C     FROM CARDS C.2.B AND C.4.                                          CHGIII
C                                                                        VINPUT
   31 MATAB = -NATAB                                                     VINPUT
      READ  (5,32)  LTYPE,LFIT,NPTS,(VMEG(I),I=1,3)                      VINPUT
   32 FORMAT (3I6,22X,3F10.0)                                            VINPUT
      IF (MATAB.GT.MXTAB3) STOP 80                                       MISC
      IF (LTYPE.EQ.2.AND.LFIT.LT.1) STOP 82                              CHGIII
      IF (LTYPE.EQ.1.AND.LFIT.LT.2) STOP 83                              VEHICL
      IF  (LTYPE.GT.0)  GO TO 34                                         VINPUT
      READ (5,33) ((ATAB(I,J),I=1,3),(ATAB(I,J),I=10,12),J=1,MATAB)      VINPUT
   33 FORMAT (10X,6F10.0)                                                VINPUT
      ISKIP = 0                                                          VINPUT
      GO TO 46                                                           VINPUT
C                                                                        CHGIII
C     FOR SPLINE FIT VEHICLE MOTION                                      CHGIII
C     READ DATA FROM CARDS C.5.                                          CHGIII
C                                                                        CHGIII
   34 LPTS = LTYPE-1 + NPTS                                              VINPUT
      IF (NPTS.GT.MXTAB3) STOP 84                                        MISC
      READ  (5,35)  (TT(I),(XYZ(I,J),J=1,6),I=1,LPTS)                    VINPUT
   35 FORMAT (7F10.0)                                                    VINPUT
      WRITE  (6,36)  LTYPE,LFIT,NPTS                                     CHGIII
   36 FORMAT ('0 SPLINE FIT TABULAR INPUT'//                             CHGIII
     *        3X,'LTYPE =',I6,'  LFIT =',I6,'  NPTS =',I6/)              CHGIII
      IF (LTYPE.EQ.2) WRITE(6,701) UNITL,UNITT,TT(1),(XYZ(1,J),J=1,6)    CHGIII
      IF (LTYPE.EQ.3) WRITE(6,702) UNITL,UNITT,TT(1),(XYZ(1,J),J=1,6),   CHGIII
     * UNITL,UNITT,UNITT,UNITT,TT(2),(XYZ(2,J),J=1,6)                    CHGIII
  701 FORMAT(32X ,'INITIAL LINEAR POSITION (',A4,')',17X,'INITIAL ANGULACHGIII
     *R POSITION (DEG)',/,3X,'TIME(',A4,')=',F9.4,3X,2('X=',F10.3,2X,    JTF984
     * 'Y=',F10.3,2X,'Z=',F10.3,8X),/)                                   CHGIII
  702 FORMAT(32X, 'INITIAL LINEAR POSITION (',A4,')',17X,'INITIAL ANGULACHGIII
     *R POSITION (DEG)',/,3X,'TIME(',A4,')=',F9.4,3X,2('X=',F10.3,2X,    JTF984
     * 'Y=',F10.3,2X,'Z=',F10.3,8X),/,30X,  'INITIAL LINEAR VELOCITY (',CHGIII
     * A4,'/',A4,')',12X,'INITIAL ANGULAR VELOCITY (DEG/',A4,')',        CHGIII
     */,3X,'TIME(',A4,')=',F9.4,3X,2('X=',F10.2,2X,'Y=',                 CHGIII
     * F10.2,2X,'Z=',F10.2,8X),/)                                        CHGIII
      IF (LTYPE.EQ.1) WRITE(6,703) UNITL,UNITT                           CHGIII
      IF (LTYPE.EQ.2) WRITE(6,704) UNITL,UNITT,UNITT,UNITT               CHGIII
      IF (LTYPE.EQ.3) WRITE(6,705) UNITT,UNITT                           VEHICL
  703 FORMAT(29X   ,'LINEAR POSITION (',A4,')',21X,'ANGULAR POSITION (DECHGIII
     *G)', /,5X,'TIME(',A4,')',11X,'X',11X,'Y',11X,'Z',18X,'YAW',8X,     VEHICL
     *'PITCH',8X,'ROLL')                                                 VEHICL
  704 FORMAT(26X,'LINEAR VELOCITY (',A4,'/',A4,')',16X,                  CHGIII
     * 'ANGULAR VELOCITY (DEG/',A4,')',/,5X,'TIME(',A4,')',              CHGIII
     * 11X,2('X',11X,'Y',11X,'Z',19X))                                   CHGIII
  705 FORMAT(26X,'LINEAR DECELERATION (G''S)',15X,                       VEHICL
     * 'ANGULAR ACCELERATION (DEG/',A4,'**2)',/,5X,'TIME(',A4,')',        CHGIII
     * 11X,2('X',11X,'Y',11X,'Z',19X))                                   CHGIII
```

```
      IF (LTYPE.EQ.1) WRITE(6,706) (TT(I),(XYZ(I,J),J=1,6),I=1,LPTS)    CHGIII
      IF (LTYPE.EQ.2) WRITE(6,706) (TT(I),(XYZ(I,J),J=1,6),I=2,LPTS)    CHGIII
      IF (LTYPE.EQ.3) WRITE(6,706) (TT(I),(XYZ(I,J),J=1,6),I=3,LPTS)    CHGIII
  706 FORMAT(1X,F12.5,6X,3F12.3,8X,3F12.3)                              CHGIII
      DO  37  I=1,3                                                     VINPUT
      XO(I) = XYZ(1,I)                                                  VINPUT
      XDOTO(I) = XYZ(2,I)                                               VINPUT
      VMEG(I) = XYZ(2,I+3)                                              VINPUT
   37 ANGLE(I) = XYZ(1,I+3)                                             JTF984
       IMJ = 6                                                          JTF984
       IF(LTYPE.EQ.1)IMJ = 3                                            JTF984
      DO  45  II=1,IMJ                                                  JTF984
      CALL SPLINE (TT(LTYPE),XYZ(LTYPE,II),F,NPTS,LFIT)                 VINPUT
      I = II                                                            VINPUT
      IF  (II.GT.3)  I = II + 6                                         VINPUT
      IF(LTYPE.EQ.1) XDOTO(I) = F(3,1)                                  JTF984
      UNITS = 1.0                                                       JTF984
      IF  (LTYPE.LT.3 .AND. II.LE.3)  UNITS = -1.0/G                    VINPUT
      K1 = 1                                                            VINPUT
      DO  45  J=1,MATAB                                                 VINPUT
      TTT = ATO + DFLOAT(J-1)*ADT                                       VINPUT
      DO  39  L=K1,NPTS                                                 JTF984
       K = L                                                            JTF984
      IF  (TTT.LT.F(1,L+1))  GO TO 40                                   VINPUT
   39 CONTINUE                                                          VINPUT
   40 K1 = K                                                            VINPUT
      DX = TTT - F(1,K)                                                 VINPUT
      IF (LTYPE-2)  41,42,43                                            BUTLER1
   41 ACC = 2.0*F(4,K) + 6.0*DX*F(5,K)                                  VINPUT
      GO TO 44                                                          VINPUT
   42 ACC = F(3,K) + DX*(2.0*F(4,K)+3.0*DX*F(5,K))                      VINPUT
      GO TO 44                                                          VINPUT
   43 ACC = F(2,K) + DX*(F(3,K)+DX*(F(4,K)+DX*F(5,K)))                  VINPUT
   44 ATAB(I,J) = ACC*UNITS                                             VINPUT
   45 CONTINUE                                                          VINPUT
      ISKIP = 1                                                         VINPUT
       IF(LTYPE.NE.1)GO TO 46                                           JTF984
C     CODE FOR OMEGA ROUTINE: COMPUTE ATAB(I,J),I=10,11,12 J = 1,MATAB  JTF984
      DO 80 I = 1,NPTS                                                  JTF984
       DO 91 K = 1,3                                                    JTF984
   91  A1(K) = XYZ(I,K+3)                                               JTF984
      CALL QUAT(A1,W1)                                                  JTF984
       DO 76 K = 1,4                                                    JTF984
   76 Q1(I,K) = W1(K)                                                   JTF984
      IF(I.EQ.1)GO TO 80                                                JTF984
       TA = 0.0                                                         JTF984
       TB = 0.0                                                         JTF984
      DO 77 K = 1,4                                                     JTF984
      TA = TA + DABS(Q1(I,K) - Q1(I-1,K))                               JTF984
   77 TB = TB + DABS(Q1(I,K) + Q1(I-1,K))                               JTF984
```

```
      IF(TA.LE.TB)GO TO 80                                              JTF984
      DO 78 K = 1,4                                                     JTF984
   78 Q1(I,K) = -Q1(I,K)                                                JTF984
   80 CONTINUE                                                          JTF984
      DO 82 K = 1,4                                                     JTF984
   82 CALL SPLINE(TT,Q1(1,K),SP(1,1,K),NPTS,LFIT)                       JTF984
      DO 90 J = 1,MATAB                                                 JTF984
      TTT = AT0 + DFLOAT(J-1)*ADT                                       JTF984
      K1 = 1                                                            JTF984
      DO 83 L = K1,NPTS                                                 JTF984
      K = L                                                             JTF984
   83 IF(TTT.LT.SP(1,L+1,1))GO TO 84                                    JTF984
   84 K1 = K                                                            JTF984
      DX = TTT - SP(1,K,1)                                              JTF984
      DO 85 L = 1,4                                                     JTF984
      W1(L) = SP(2,K,L)+DX*(SP(3,K,L)+DX*(SP(4,K,L)+DX*SP(5,K,L)))      JTF984
      QD(L) = 2.0*SP(4,K,L) + 6.0*DX*SP(5,K,L)                          JTF984
   85 IF(J.EQ.1)QC(L) = SP(3,K,L)+DX*(2.0*SP(4,K,L)+DX*3.0*SP(5,K,L))   MISC
      CCC = 2.0/RADIAN                                                  JTF984
      IF(J.GT.1)GO TO 88                                                JTF984
     CALL CROSS(QC(2),W1(2),A1)                                         JTF984
      DO 86 K = 1,3                                                     JTF984
   86 VMEG(K) = CCC*(W1(1)*QC(K+1) - QC(1)*W1(K+1) + A1(K))             JTF984
      CALL DRCQUA(DVEH,W1)                                              JTF984
      CALL YPRDEG(DVEH,ANGLE)                                           JTF984
   88 CALL CROSS(QD(2),W1(2),QC(2))                                     JTF984
      DO 89 K = 2,4                                                     JTF984
   89 ATAB(K+8,J) = CCC*(W1(1)*QD(K)-QD(1)*W1(K) + QC(K))               JTF984
   90 CONTINUE                                                          JTF984
   46 DO  55  J=1,MATAB                                                 VINPUT
      IF (MOD(J,45).NE.1) GO TO 49                                      VINPUT
C                                                                       VINPUT
C     PRINT PAGE HEADING AT START OF EACH 45 TIME POINTS.               VINPUT
C                                                                       VINPUT
      IPAGE = (J-1)/45 + 1                                              VINPUT
      IF (ISKIP.EQ.1) WRITE (6,75) NPG                                  PAGE
      IF (ISKIP.EQ.1) NPG=NPG+1                                         PAGE
   75 FORMAT('1',122X,'PAGE',I5)                                        PAGE
      WRITE (6,48) VPSTTL,IPAGE,UNITL,UNITT,UNITL                       PAGE
   48 FORMAT('0  VEHICLE LINEAR TIME HISTORY',3X,20A4,3X,               PAGE
     *             'PAGE NO.',I3//                                      VINPUT
     *       4X,'TIME',12X,'LINEAR DECELERATIONS (G''S)',               VINPUT
     *                   11X,'LINEAR VELOCITIES (',A4,'/',A4,')',       VINPUT
     *                   11X,'LINEAR DISPLACEMENTS (',A4,')'     /      VINPUT
     *         3X,'(MSEC)',3(11X,'X',11X,'Y',11X,'Z',3X)       /  )     VINPUT
      ISKIP = 1                                                         VINPUT
   49 IF (J.GT.1) GO TO 52                                              VINPUT
C                                                                       VINPUT
C     INTEGRATION INITIALIZATION FOR TIME = 0.                          VINPUT
C                                                                       VINPUT
```

```
      DO 50 I=1,3                                                       VINPUT
      ATAB(I+6,J) = XO(I)                                               VINPUT
   50 ATAB(I+12,J) = VMEG(I)                                            JTF984
      CALL DRCYPR (DVEH,ANGLE,IDYPR)                                    VINPUT
      DO 51 I=1,3                                                       VINPUT
      IF (LTYPE.EQ.0)  XDOTO(I) = VIPS*DVEH(1,I)                        VINPUT
   51 ATAB(I+3,J) = XDOTO(I)                                            VINPUT
      GO TO 54                                                          VINPUT
   52 DO 53 I=1,3                                                       VINPUT
C                                                                       VINPUT
C     INTEGRATE LINEAR VELOCITY AND DISPLACEMENT.                       VINPUT
C                                                                       VINPUT
      ATAB(I+3,J) = ATAB(I+3,J-1)-G*ADT/2.0*(ATAB(I,J-1)+ATAB(I,J))     VINPUT
   53 ATAB(I+6,J) = ATAB(I+6,J-1)                                       VINPUT
     *     +ADT*(ATAB(I+3,J-1)-G*ADT/6.0*(2.0*ATAB(I,J-1)+ATAB(I,J)))   VINPUT
   54 T1 = (ATO + DFLOAT(J-1)*ADT)*1000.0                               VINPUT
   55 WRITE(6,56) T1,(ATAB(I,J),I=1,9)                                  VINPUT
   56 FORMAT(F9.3,3(3X,3F12.3))                                         VINPUT
      DO 61 J=1,MATAB                                                   VINPUT
      IF (MOD(J,45).NE.1) GO TO 58                                      VINPUT
C                                                                       VINPUT
C     PRINT PAGE HEADING AT START OF EACH 45 TIME POINTS.               VINPUT
C                                                                       VINPUT
      IPAGE = (J-1)/45 + 1                                              VINPUT
      WRITE (6,57) VPSTTL,NPG,IPAGE,UNITT,UNITT                         PAGE
      NPG=NPG+1                                                         PAGE
   57 FORMAT('1 VEHICLE ANGULAR TIME HISTORY',3X,20A4,10X,'PAGE',I5/    PAGE
     *        116X,'PAGE NO.',I3/                                       PAGE
     *         4X,'TIME', 7X,'ANGULAR ACCELERATIONS (DEG/',A4,'**2)',   VINPUT
     *                    7X,'ANGULAR VELOCITIES (DEG/',A4,')',         VINPUT
     *                   11X,'ANGULAR DISPLACEMENTS (DEG)' /            VINPUT
     *       3X,'(MSEC)',2(11X,'X',11X,'Y',11X,'Z',3X),                 VINPUT
     *        10X,'YAW',8X,'PITCH',8X,'ROLL' /)                         VINPUT
   58 IF(J.EQ.1) GO TO 60                                               VINPUT
C                                                                       VINPUT
C     INTEGRATE ANGULAR VELOCITY AND DISPLACEMENT.                      VINPUT
C                                                                       VINPUT
      DO 59 I=1,3                                                       VINPUT
      ATAB(I+12,J) = ATAB(I+12,J-1)+(ATAB(I+9,J-1)+ATAB(I+9,J))*ADT/2.0 VINPUT
   59 THET(I) = ADT*(ATAB(I+12,J-1)+(2.0*ATAB(I+9,J-1)+ATAB(I+9,J))*ADT VINPUT
     */6.0)*RADIAN                                                      VINPUT
      CALL DSETD(DVEH,THET,THT)                                         VINPUT
   60 CALL YPRDEG(DVEH,THET)                                            VINPUT
      T1 = (ATO + DFLOAT(J-1)*ADT)*1000.0                               VINPUT
   61 WRITE (6,56)  T1,(ATAB(I,J),I=10,15),THET                         VINPUT
C                                                                       VINPUT
C     PROGRAM INITIALIZATION FOR TIME = 0.                              VINPUT
C                                                                       VINPUT
      CALL DRCYPR (DVEH,ANGLE,IDYPR)                                    VINPUT
      DO 63 I=1,3                                                       VINPUT
```

```
      XACOMP(I) = -G*ATAB(I,1)                                         VINPUT
      VMEG(I)   = ATAB(I+12,1)*RADIAN                                  VINPUT
   63 VMEGD(I) = ATAB(I+9 ,1)*RADIAN                                   VINPUT
   64 J = MSEG                                                         VINPUT
      IF  (MSEG.EQ.0)  GO TO 65                                        VINPUT
      IF  (MSEG.LE.NSEG)  GO TO 66                                     VINPUT
      IF  (MSEG.NE.NVEH+1)  STOP 6                                     VINPUT
   65 NVEH = NVEH+1                                                    VINPUT
      J = NVEH                                                         VINPUT
C                                                                      VINPUT
C     SETUP FOR ALL PRESCRIBED SEGMENT MOTION.                         VINPUT
C                                                                      VINPUT
   66 NVH = NVH+1                                                      VINPUT
      ISING(J) = -1                                                    VINPUT
      IF  (MSEG.GT.NSEG)  SEG(J) = VEH(NVH)                            VINPUT
      RW(J) = 0.0                                                      VINPUT
      DO  67  I=1,3                                                    VINPUT
      RPHI (I,J) = 0.0                                                 VINPUT
      SEGLA(I,J) = VMEGD(I)                                            VINPUT
      WMEGD(I,J) = XACOMP(I)                                           VINPUT
   67 AXV(I,NVH) = AX(I)                                               VINPUT
      VTO(NVH) = ATO                                                   VINPUT
      VDT(NVH) = ADT                                                   VINPUT
      OMEGV(NVH) = OMEG                                                VINPUT
      TIMEV(NVH) = VTIME                                               VINPUT
      NVTAB(NVH) = NATAB                                               VINPUT
      INDXV(NVH) = J                                                   VINPUT
      NJ = IABS(NATAB)                                                 VINPUT
      IF  (NJ.LE.0)  GO TO 69                                          VINPUT
      DO  68  K=1,NJ                                                   VINPUT
      DO  68  I=1,3                                                    VINPUT
      VATAB(I   ,K,NVH) = ATAB(I,K)                                    VINPUT
   68 VATAB(I+3,K,NVH) = ATAB(I+9,K)                                   VINPUT
   69 IF  (J.LE.NSEG)  GO TO 72                                        VINPUT
C                                                                      VINPUT
C     SETUP FOR NEW VEHICLE (SEGMENT) MOTION.                          VINPUT
C                                                                      VINPUT
      W(J)  = 0.0                                                      VINPUT
      RW(J) = 0.0                                                      VINPUT
      DO  71  I=1,3                                                    VINPUT
      DO  70  K=1,3                                                    VINPUT
      D(I,K,J) = DVEH(I,K)                                             VINPUT
   70 SGTEST(I,K,J) = 0.0                                              VINPUT
      SGTEST(I,4,J) = 0.0                                              VINPUT
      SEGLP(I,J) = XO(I)                                               VINPUT
      SEGLV(I,J) = XDOTO(I)                                            VINPUT
      WMEG (I,J) = VMEG(I)                                             VINPUT
      PHI  (I,J) = 0.0                                                 VINPUT
   71 RPHI (I,J) = 0.0                                                 VINPUT
   72 IF  (MSEG.NE.0)  GO TO 12                                        VINPUT
```

```
      SEG(NVEH) = VEH(6)                                              VINPUT
C                                                                     VINPUT
C     SET UP SEGMENT DATA FOR GROUND                                  VINPUT
C                                                                     VINPUT
      NGRND = NVEH+1                                                  VINPUT
      IF  (NGRND.GT.30 .OR. NVH.GT.6)  STOP 7                         VINPUT
      SEG(NGRND) = GRND                                               VINPUT
      J = NGRND                                                       VINPUT
      ISING(J) = -1                                                   VINPUT
      W(J)   = 0.0                                                    VINPUT
      RW(J) = 0.0                                                     VINPUT
      DO  74  I=1,3                                                   VINPUT
      DO  73  K=1,3                                                   VINPUT
      D(I,K,J) = 0.0                                                  VINPUT
   73 SGTEST(I,K,J) = 0.0                                             VINPUT
      D(I,I,J) = 1.0                                                  VINPUT
      SGTEST(I,4,J) = 0.0                                             VINPUT
      SEGLP(I,J) = 0.0                                                VINPUT
      SEGLV(I,J) = 0.0                                                VINPUT
      SEGLA(I,J) = 0.0                                                VINPUT
      WMEG (I,J) = 0.0                                                VINPUT
      WMEGD(I,J) = 0.0                                                VINPUT
      PHI  (I,J) = 0.0                                                VINPUT
   74 RPHI (I,J) = 0.0                                                VINPUT
      RETURN                                                          VINPUT
      END                                                             VINPUT
```

```
      DOUBLE PRECISION FUNCTION VISCOS(ZD,VISC,HA)                      VISCOS
C                                                 REV 19    10/23/78VISCOS
C     COMPUTES SUM OF COULOMB AND VISCOUS TORQUES                       VISCOS
C     AT JOINTS AS A FUNCTION OF THETA DOT.                             VISCOS
C     ACTUALLY ROUTINE RETURNS SUM/ZD.                                  VISCOS
C                                                                       VISCOS
C         ARGUMENTS:                                                    VISCOS
C          ZD  : !THETA DOT! WHERE THETA IS THE ANGLE OF THE JOINT.     VISCOS
C           VISC: ARRAY OF 5 VALUES DESCRIBING FUNCTION EVALUATION.     VISCOS
C                                                                       VISCOS
      IMPLICIT REAL*8 (A-H,O-Z)                                         VISCOS
      DIMENSION VISC(5)                                                 VISCOS
      Z = ZD                                                            VISCOS
      IF (ZD.LT.VISC(3)) Z = VISC(3)/(2.0-ZD/VISC(3))                   VISCOS
      HA = (Z-ZD)/Z                                                     VISCOS
      VISCOS = VISC(1)+VISC(2)/Z                                        VISCOS
      RETURN                                                            VISCOS
      END                                                               VISCOS
```

```
      SUBROUTINE VISPR(IJ,NJ)                                          VISPR
C                                                 REV IV    02/01/88MISDOT
C     COMPUTES VISCOS AND SPRING TORQUES AT THE JOINTS                 VISPR
C     AND ADDS THEM TO THE U2 ARRAY.                                   VISPR
C                                                                      VISPR
C     ARGUMENTS:                                                       VISPR
C        NJ = 0 - REGULAR COMPUTATION FOR ALL JOINTS                   VISPR
C           # 0 - COMPUTE ONLY FOR JOINT NJ IMPULSE                    VISPR
C                                                                      VISPR
C        IJ = 1 IMPULSE FOR FLEXURE ONLY                               VISPR
C           = 2 IMPULSE FOR TORSION ONLY                               VISPR
C           = 4 IMPULSE FOR GLOBALGRAPHIC ONLY                         VISPR
C                                                                      VISPR
      IMPLICIT REAL*8 (A-H,O-Z)                                        VISPR
      COMMON/CONTRL/ TIME,NSEG,NJNT,NPL,NBLT,NBAG,NVEH,NGRND,          VISPR
     *               NS,NQ,NSD,NFLX,NHRNSS,NWINDF,NJNTF,NPRT(36),NPG   PAGE
      COMMON/SGMNTS/ D(3,3,30),WMEG(3,30),WMEGD(3,30),U1(3,30),U2(3,30),VISPR
     *               SEGLP(3,30),SEGLV(3,30),SEGLA(3,30),NSYM(30)      VISPR
      COMMON/DESCRP/ PHI(3,30),W(30),RW(30),SR(4,60),HA(3,60),HB(3,60), SLIP
     *               RPHI(3,30),HT(3,3,60),SPRING(5,90),VISC(7,90),    VISPR
     *               JNT(30),IPIN(30),ISING(30),IGLOB(30),JOINTF(30)   VISPR
      COMMON/CMATRX/ V1(3,30),V2(3,30),V3(3,12),B12(3,3,60),A22(3,3,60),VISPR
     *               F(3,30),TQ(3,30),WJ(30),A11(3,3,30)               SLIP
      COMMON/FORCES/PSF(7,70),BSF(4,20),SSF(10,40),BAGSF(3,20),        NCFORC
     *               PRJNT(7,30),NPANEL(5),NPSF,NBSF,NSSF,NBGSF        VISPR
      COMMON/CEULER/ IEULER(30),HIR(3,3,90),ANG(3,30),ANGD(3,30),      JDRIFT
     *               FE(3,30),TQE(3,30),CONST(5,30)                    JDRIFT
      COMMON/TEMPVI/ CREST,TTI(3),R1I(3),R2I(3),JSTOP(4,2,30)          VISPR
      COMMON/CNSNTS/ PI,RADIAN,G,THIRD,EPS(24),                        VISPR
     *               UNITL,UNITM,UNITT,GRAVTY(3),TWOPI                 TWOPI
      COMMON/TEMPVS/ T3(3),T6(3),T7(3),T8(3),T9(3),                    VISPR
     *               WIJ(3),ANGL(3),DH1(3,3),HD3(3,3),                 VISPR
     *               HAD,HBD,WIJM,CV,CSA,CSB,TQC                       VISPR
      IF (NJNT.LE.0) GO TO 99                                          VISPR
      CALL ELTIME(1,13)                                                VISPR
      IF (NPRT(12).NE.0) WRITE (6,11)  TIME,NPG                        PAGE
      IF (NPRT(12).NE.0) NPG=NPG+1                                     PAGE
   11 FORMAT('1 VISPR COMPUTATIONS FOR TIME =',F12.6,80X,'PAGE',I5)    PAGE
      J1 = 1                                                           VISPR
      J2 = NJNT                                                        VISPR
      IF (NJ.EQ.0) GO TO 13                                            VISPR
      J1 = NJ                                                          VISPR
      J2 = NJ                                                          VISPR
   13 DO 90 J=J1,J2                                                    VISPR
      DO 12 L=1,3                                                      VISPR
      T3(L) = 0.0                                                      VISPR
      T6(L) = 0.0                                                      VISPR
      ANGL(L) = 0.0                                                    VISPR
   12 TQ(L,J) = 0.0                                                    VISPR
      WJ(J) = 0.0                                                      VISPR
```

```
C                                                                       VISPR
C     DO NOT COMPUTE TORQUES FOR NULL, LOCKED OR EULER JOINTS.          VISPR
C                                                                       VISPR
      I = IABS(JNT(J))                                                  VISPR
      IF (I.LE.0) GO TO 90                                              VISPR
      CALL DOT33 (D(1,1,J+1),HT(1,1,2*J),HIR(1,1,J))                    VISPR
      IF  (IABS(IPIN(J)).EQ.4)  GO TO 90                                SLIP
C                                                                       VISPR
C     ZERO T1-T9 ARRAYS AND HAD,HBD,WIJM,CV,CS4,CSB AND TQC.            VISPR
C                                                                       VISPR
      WIJM = 0.0                                                        VISPR
      HAC = 0.0                                                         BUTLER1
      CV   = 0.0                                                        VISPR
      CSA  = 0.0                                                        VISPR
      CSB  = 0.0                                                        VISPR
      TQC  = 0.0                                                        VISPR
      CALL DOT33 (D(1,1,I),HT(1,1,2*J-1),DH1)                           VISPR
      CALL DOT33 (DH1,HIR(1,1,J),HD3)                                   VISPR
      DO 220 L=1,3                                                      TGMOD6
      DO 220 K=1,3                                                      TGMOD6
      IF(DABS(HD3(L,K)).LT.EPS(10)) HD3(L,K) = 0.D0                     TGMOD6
  220 CONTINUE                                                          TGMOD6
      HAD  = HD3(3,3)                                                   VISPR
      IF  (HAD.GT. 1.0)  HAD =  1.0                                     VISPR
      IF  (HAD.LT.-1.0)  HAD = -1.0                                     VISPR
      ANGL(1) = DACOS(HAD)                                              VISPR
      IF ((HD3(2,3).NE.0.0 .OR. HD3(1,3).NE.0.0).AND.IABS(IPIN(J)).NE.7)SLIP
     *ANGL(2) = DATAN2(HD3(2,3),HD3(1,3))                               VISPR
      ANGL(3) = DATAN2(HD3(2,1)-HD3(1,2),HD3(1,1)+HD3(2,2))             VISPR
      IF(NPRT(12).NE.0.AND.IPIN(J).LT.0) WRITE (6,739) J,I,ANGL,        TGMOD6
     *((D(L,K,J+1),K=1,3),(HT(L,K,2*J),K=1,3),(HIR(L,K,J),K=1,3),L=1,3),TGMOD6
     *((D(L,K,I),K=1,3),(HT(L,K,2*J-1),K=1,3),(DH1(L,K),K=1,3),L=1,3),  TGMOD6
     *          ((HD3(L,K),K=1,3),L=1,3)                                TGMOD6
  739 FORMAT(1H0,'J= ',I2,1X,'I= ',I2,3(2X,D14.7),/,                    TGMOD6
     *      2(3(9(1X,D13.6),/),/),3(3(2X,D18.12),/))                    TGMOD6
      IF  (IPIN(J).LT.0)  GO TO 41                                      VISPR
      IF (NJ.NE.0.AND.IJ.EQ.4) GO TO 27                                 VISPR
C                                                                       VISPR
C     CONVERT TO INERTIAL REFERENCE SYSTEM                              VISPR
C        T1= D(I)'*HA(NJ)      T4=D(J+1)'*HA(MJ)                        VISPR
C        T3= D(I)'*WMEG(I)     T6=D(J+1)'*WMEG(J+1)                     VISPR
C                                                                       VISPR
C     HAD = COS TA = T1.T4                                              VISPR
C     WIJ = T3-T6                                                       VISPR
C     WJ  = !WIJ!                                                       VISPR
C                                                                       VISPR
      DO 20 L=1,3                                                       VISPR
      DO 15 M=1,3                                                       VISPR
      T3(L) = T3(L)+ D(M,L,I)* WMEG(M,I)                                VISPR
   15 T6(L) = T6(L)+ D(M,L,J+1)* WMEG(M,J+1)                            VISPR
```

```
      WIJ(L)= T3(L)-T6(L)                                                   VISPR
   20 WIJM = WIJM + WIJ(L)**2                                               VISPR
      WIJM = DSQRT(WIJM)                                                    VISPR
      IF (WIJM.LE.EPS(12)) WIJM = 0.0                                       MISDOT
      WJ(J) = WIJM                                                          VISPR
C                                                                           VISPR
C     T7 = T1 X T4                                                          VISPR
C     HAC = !T7!                                                            VISPR
C                                                                           VISPR
      CALL CROSS (DH1(1,3),HIR(1,3,J),T7)                                   VISPR
      HACC = T7(1)**2 + T7(2)**2 + T7(3)**2                                 VISPR
      HAC = DSQRT(HACC)                                                     VISPR
C                                                                           VISPR
C     COMPUTE CV, THE MAGNITUDE OF VISCOUS AND COULOMB TORQUE/WIJM          VISPR
C             RA = +SGN TA DOT = -WIJ.T7                                    VISPR
C        AND CSA, THE MAGNITUDE OF FLEXURE TORQUE/HAC                       VISPR
C                                                                           VISPR
      CV = VISCOS(WIJM,VISC(1,3*J-2),HA2)                                   VISPR
      IF  (NJ.EQ.0)  HA(2,2*J) = HA2                                        VISPR
      CREST = VISC(7,3*J-2)                                                 VISPR
      RA = -(WIJ(1)*T7(1) + WIJ(2)*T7(2) + WIJ(3)*T7(3))                    VISPR
      IF (HAC.LT.EPS(12)) RA=0.0                                            MISDOT
      IF (HAC.GE.EPS(12)) RA=RA/HAC                                         MISDOT
      JSTP = 0                                                              VISPR
      IF (IPIN(J).EQ.7) GOTO 25                                             SLIP
      IF (JOINTF(J).EQ.0) CSA = EFUNCT(ANGL(1),RA,SPRING(1,3*J-2),JSTP)     VISPR
      IF (JOINTF(J).NE.0) CSA = FNTERP(ANGL(1),ANGL(2),JOINTF(J))           VISPR
      IF (HAC.LT.EPS(12)) CSA=0.0                                           MISDOT
      IF (HAC.GE.EPS(12)) CSA=CSA/HAC                                       MISDOT
25    IF (NJ.EQ.0) JSTOP(1,1,J) = JSTP                                      SLIP
      IF (IPIN(J).EQ.1) GO TO 34                                            VISPR
      IF (IPIN(J).EQ.6) GOTO 34                                             SLIP
C                                                                           VISPR
C     RB = +SGN TB DOT = -WIJ.T8                                            VISPR
C     COMPUTE CSB, THE MAGNITUDE OF TORSIONAL TORQUE/HBC                    VISPR
C                                                                           VISPR
      RB  = -(WIJ(1)*HIR(1,3,J) + WIJ(2)*HIR(2,3,J) + WIJ(3)*HIR(3,3,J))VISPR
      CSB = EFUNCT(ANGL(3),RB,SPRING(1,3*J-1),JSTP)                         VISPR
      IF (NJ.EQ.0) JSTOP(2,1,J) = JSTP                                      VISPR
      IF (NJ.GT.0) GO TO 34                                                 VISPR
C                                                                           VISPR
C     COMPUTE EFFECT OF GLOBALGRAPHIC JOINT STOP (IPIN=3)                   VISPR
C                                                                           VISPR
   27 IF (IPIN(J).NE.3) GO TO 34                                            VISPR
      CALL GLOBAL (J,HD3(1,3),DH1,TQC,T9,ANGL)                              VISPR
C                                                                           VISPR
C     COMPUTE TOTAL TORQUE IN INERTIAL REFERENCE BY                         VISPR
C     TQ = -CV*WIJ + CSA*T7 + CSB*T8 + TQC*T9                               VISPR
C                                                                           VISPR
   34 IF (NJ.EQ.0) GO TO 35                                                 JDRIFT
```

```
      CV = 0.0                                                          VISPR
      IF (IJ.NE.1) CSA = 0.0                                            VISPR
      IF (IJ.NE.2) CSB = 0.0                                            VISPR
      IF (IJ.NE.4) TQC = 0.0                                            VISPR
  35  IF (HA(2,2*J).EQ.0.0) GO TO 36                                    JDRIFT
      CALL MAT31 (HIR(1,1,J),HA(1,2*J-1),TQ(1,J))                       VISPR
      DO 38 L=1,3                                                       VISPR
   38 TQ(L,J) = HA(2,2*J)*TQ(L,J)                                       VISPR
   36 DO 37 L=1,3                                                       VISPR
      TQ(L,J) = TQ(L,J) -CV*WIJ(L) +CSA*T7(L) +CSB*HIR(L,3,J) +TQC*T9(L)VISPR
   37 TTI(L) = TQ(L,J)                                                  VISPR
      IF (NPRT(12).NE.0) WRITE (6,39)                                   VISPR
     *         J,CV,CSA,CSB,HAC,RA,RB,(TQ(L,J),L=1,3),                  VISPR
     *         WIJ,T7,ANGL, DH1 , HD3,                                  VISPR
     *         ((HIR(L,K,J),L=1,3),K=1,3)                               VISPR
   39 FORMAT (1H0,I3,3F14.3,6F14.6/(4X,9F14.6))                         VISPR
C                                                                       VISPR
C     ADD TORQUE CONVERTED TO LOCAL REFERENCE BY                        VISPR
C      U2I = U2I + DI*TQ                                                VISPR
C      U2J = U2J - DJ*TQ                                                VISPR
C                                                                       VISPR
      DO 40 L=1,3                                                       VISPR
      DO 40 M=1,3                                                       VISPR
      U2(L,I  ) = U2(L,I  ) + D(L,M,I  )*TQ(M,J)                        VISPR
   40 U2(L,J+1) = U2(L,J+1) - D(L,M,J+1)*TQ(M,J)                        VISPR
C                                                                       VISPR
C     STORE DATA FOR OUTPUT ROUTINE INTO PRJNT ARRAY.                   VISPR
C                                                                       VISPR
   41 PRJNT(1,J) = IPIN(J)                                              VISPR
      PRJNT(2,J) = ANGL(1)                                              VISPR
      PRJNT(3,J) = ANGL(2)                                              VISPR
      PRJNT(4,J) = ANGL(3)                                              VISPR
      PRJNT(5,J) = (CSA*HAC)**2 + CSB**2                                VISPR
      PRJNT(6,J) = (CV*WIJM)**2                                         VISPR
      PRJNT(7,J) = TQ(1,J)**2 + TQ(2,J)**2 + TQ(3,J)**2                 VISPR
   90 CONTINUE                                                          VISPR
      CALL ELTIME(2,13)                                                 VISPR
   99 RETURN                                                            VISPR
      END                                                               VISPR
```

```
      SUBROUTINE WINDY(MMM,MM,N,NN,NT)                                  WINDY
C                                                REV IV    07/23/86TWOPI
C     COMPUTES FORCES AND TORQUES ADDING THEM TO THE U1 AND U2 ARRAYS   WINDY
C     OF WIND BLAST FORCES DETERMINED BY FUNCTION STORED IN TAB(NT)     WINDY
C     ON ELLIPSOID (MM) ATTACHED TO BODY SEGMENT (M) WHICH EXTENDS      WINDY
C     THROUGH THE INTERSECTING PLANE (NN) ATTACHED TO SEGMENT (N).      WINDY
C                                                                       WINDY
       IMPLICIT REAL*8 (A-H,O-Z)                                        WINDY
      COMMON/CONTRL/ TIME,NSEG,NJNT,NPL,NBLT,NBAG,NVEH,NGRND,           WINDY
     *               NS,NQ,NSD,NFLX,NHRNSS,NWINDF,NJNTF,NPRT(36),NPG    PAGE
      COMMON/SGMNTS/ D(3,3,30),WMEG(3,30),WMEGD(3,30),U1(3,30),U2(3,30),WINDY
     *               SEGLP(3,30),SEGLV(3,30),SEGLA(3,30),NSYM(30)       WINDY
      COMMON/TABLES/MXNTI,MXNTB,MXTB1,MXTB2,NTI(50),NTAB(1250),TAB(4500)DIMENB
      COMMON/WINDFR/ WTIME(30),QFU(3,5),QFV(3,5),WF(3,30),IWIND(30),    WINDOP
     *               MWSEG(7,30),NFVSEG(6),NFVNT(5),MOWSEG(30,30)       WINDOP
      COMMON/CNTSRF/ PL(24,30),BELT(20,8),TPTS(6,8),BD(24,40)           EDGE
      COMMON/CNSNTS/ PI,RADIAN,G,THIRD,EPS(24),                         WINDY
     *               UNITL,UNITM,UNITT,GRAVTY(3),TWOPI                  TWOPI
      COMMON/TEMPVS/ DMNT(3,3),XMN(3),XMM(3),TM(3),BET,BTS,P,FT(3),     WINDOP
     *               FF(3),AF(3),FAF,TF,BREF,SCALE,TRACER,AREA,RLM(3),  WINDOP
     *               TQM(3),RM(3),DD(3,3),DDD(3,3),R(3,3),DVP(3,3),     WINDOP
     *               SI(3,15),R2(2,3),TTF(3),FFT(3),AM(3,3),VP(3),      WINDOP
     *               SS(3),SM(3),SN1(3),AS(3),BTE,XNORM,TEMP,           WINDOP
     *               X,Y,AI(3,3,15),RYC,AMDA1,AMDA2,B1,B2,RXC           WINDOP
C                                                                       WINDOP
C     MMM=0   CALCULATE NFORCE                                          WINDOP
C     MMM>0   WIND FORCE CALCULATED USING ENTIRE AREA METHOD            WINDOP
C     MMM<0   WIND FORCE CALCULATED USING GRID METHOD                   WINDOP
C                     (ALLOWS BLOCKING SEGMENTS)                        WINDOP
C                                                                       WINDOP
      DATA NSTEPS/4/                                                    WINDOP
      CALL ELTIME(1,37)                                                 WINDY
      M=IABS(MMM)                                                       WINDOP
      IF (MMM.EQ.0) GO TO 50                                            WINDOP
C                                                                       WINDY
C     COMPUTE PENETRATION DISTANCE; IF NEGATIVE, RETURN.                WINDY
C                                                                       WINDY
      CALL DOTT33 (D(1,1,M),D(1,1,N),DMNT)                              WINDY
      DO 10 I=1,3                                                       WINDY
   10 XMN(I) = SEGLP(I,M) - SEGLP(I,N)                                  WINDY
      CALL MAT31 (D(1,1,M),XMN,XMM)                                     WINDY
      CALL MAT31 (DMNT,PL(1,NN),TM)                                     WINDY
      BET = PL(4,NN)                                                    WINDY
      DO 11 I=1,3                                                       WINDY
   11 BET = BET - TM(I)*(BD(I+3,MM)+XMM(I))                             WINDY
      CALL MAT31 (BD(16,MM),TM,RM)                                      WINDY
      BTS = TM(1)*RM(1) + TM(2)*RM(2) + TM(3)*RM(3)                     WINDY
      BTE = -DSQRT(BTS)                                                 WINDY
      P   = BET - BTE                                                   WINDY
      IF (P.LT.0.0) GO TO 99                                            WINDY
```

```
C                                                                      WINDY
C     FETCH OR STORE INITIAL PENETRATION TIME.                         WINDY
C                                                                      WINDY
      IWIND(M) = M                                                     WINDY
      IF (TIME.LE.WTIME(M)) WTIME(M) = TIME                            WINDY
      FTIME = TIME - WTIME(M)                                          WINDY
C                                                                      WINDY
C     GET DRAG COEFFICIENT CD FROM TABLE NTC FOR TIME = FTIME.         WINDOP
C                                                                      WINDOP
      CD=1.0                                                           WINDOP
      NTC=MWSEG(6,M)                                                   WINDOP
      IF (NTC.EQ.0) GOTO 20                                            WINDOP
      KT=NTI(NTC)                                                      WINDOP
      NENTRY=TAB(KT+5)                                                 WINDOP
      K1=KT+10                                                         WINDOP
      K2=4*NENTRY+KT+2                                                 WINDOP
      IF (NENTRY.EQ.1) GOTO 18                                         WINDOP
      DO 17 K=K1,K2,4                                                  WINDOP
      IF (FTIME.GT.TAB(K)) GOTO 17                                     WINDOP
      KK=K                                                             WINDOP
      R1=(TAB(K)-FTIME)/(TAB(K)-TAB(K-4))                              WINDOP
      GOTO 19                                                          WINDOP
17    CONTINUE                                                         WINDOP
18    KK=K2                                                            WINDOP
      R1=0.0                                                           WINDOP
19    R22=1.0-R1                                                       WINDOP
      K=KK+1                                                           WINDOP
      CD=R22*TAB(K)+R1*TAB(K-4)                                        WINDOP
C                                                                      WINDOP
C     GET FORCE VECTOR FT                                              WINDOP
C                                                                      WINDOP
C          RK=0   TIME DEPENDENT WIND FORCE FROM TABLE                 WINDOP
C          RK#0   VELOCITY DEPENDENT WIND FORCE                        WINDOP
C                                                                      WINDOP
20    KT = NTI(NT)                                                     WINDOP
      RK=TAB(KT)                                                       WINDOP
      IF (RK.EQ.0.0) GOTO 13                                           WINDOP
      C=TAB(KT+1)                                                      WINDOP
      PR=TAB(KT+2)                                                     WINDOP
      NSV=IDINT(TAB(KT+3))                                             WINDOP
      NSR=IDINT(TAB(KT+4))                                             WINDOP
      DO 12 I=1,3                                                      WINDOP
      V=SEGLV(I,NSV)-SEGLV(I,NSR)                                      WINDOP
12    FT(I)=DSIGN(0.5D0,-V)*CD*RK*PR*V**2/C**2                         WINDOP
      GOTO 14                                                          WINDOP
13    NSR=IDINT(TAB(KT+4))                                             WINDOP
      NENTRY = TAB(KT+5)                                               WINDY
      K1 = KT+10                                                       WINDY
      K2 = 4*NENTRY + KT+2                                             WINDY
      IF (NENTRY.EQ.1) GO TO 31                                        WINDY
```

```
      DO 30 K=K1,K2,4                                                   WINDY
      IF (FTIME.GT.TAB(K)) GO TO 30                                     WINDY
      KK = K                                                            WINDY
      R1 = (TAB(K)-FTIME)/(TAB(K)-TAB(K-4))                             WINDY
      GO TO 32                                                          WINDY
   30 CONTINUE                                                          WINDY
   31 KK = K2                                                           WINDY
      R1 = 0.0                                                          WINDY
   32 R22= 1.0 - R1                                                     WINDOP
      DO 33 I=1,3                                                       WINDY
      K= KK+I                                                           WINDY
   33 FT(I)=(R22*TAB(K) + R1*TAB(K-4))*CD                               WINDOP
      IF (NSR.EQ.0) GOTO 14                                             WINDOP
      CALL DOT31(D(1,1,NSR),FT,FF)                                      WINDOP
      DO 21 I=1,3                                                       WINDOP
21    FT(I)=FF(I)                                                       WINDOP
14    IF (MMM.LT.0) GOTO 15                                             WINDOP
C                                                                       WINDY
C     COMPUTE PRESENTED AREA TO WIND FORCE.                             WINDY
C                                                                       WINDY
      CALL MAT31 (D(1,1,M),FT,FF)                                       WINDY
      CALL MAT31 (BD(7,MM),FF,AF)                                       WINDY
      FAF = FF(1)*AF(1) + FF(2)*AF(2) + FF(3)*AF(3)                     WINDY
      IF (FAF.LE.0.0) GO TO 99                                          WINDY
      TF  = TM(1)*FF(1) + TM(2)*FF(2) + TM(3)*FF(3)                     WINDY
      BREF=0.0                                                          CCWIND
      TEMP=BTS-TF*TF/FAF                                                CCWIND
      IF(TEMP.GT.0.0) BREF =DSQRT(TEMP)                                 CCWIND
      SCALE = (-BET+BREF)/(-BTE+BREF)                                   WINDY
      IF (SCALE.GE.1.0) GO TO 99                                        WINDY
      IF (SCALE.LT.0.0) SCALE = 0.0                                     WINDY
      TRACER = (BD( 7,MM)-AF(1)**2/FAF)*(BD(11,MM)-AF(2)**2/FAF)        WINDY
     *       + (BD( 7,MM)-AF(1)**2/FAF)*(BD(15,MM)-AF(3)**2/FAF)        WINDY
     *       + (BD(11,MM)-AF(2)**2/FAF)*(BD(15,MM)-AF(3)**2/FAF)        WINDY
     *       - (BD( 8,MM)-AF(1)*AF(2)/FAF)**2                           WINDY
     *       - (BD( 9,MM)-AF(1)*AF(3)/FAF)**2                           WINDY
     *       - (BD(12,MM)-AF(2)*AF(3)/FAF)**2                           WINDY
      AREA = (1.0-SCALE**2) * PI / DSQRT(TRACER)                        WINDY
C                                                                       WINDY
C     ADD FORCE AND TORQUES TO U1 AND U2 ARRAYS FOR SEGMENT M.          WINDY
C                                                                       WINDY
      SCALE = SCALE/BTE                                                 WINDY
      DO 36 I=1,3                                                       WINDY
      RLM(I) = RM(I)*SCALE + BD(I+3,MM)                                 WINDY
      FT (I) = FT(I)*AREA                                               WINDY
   36 FF (I) = FF(I)*AREA                                               WINDY
      CALL CROSS (RLM,FF,TQM)                                           WINDY
      DO 39 I=1,3                                                       WINDY
      WF(I,M)=FT(I)                                                     WINDOP
      U1(I,M) = U1(I,M) + FT(I)                                         WINDY
```

```
   39 U2(I,M) = U2(I,M) + TQM(I)                                   WINDY
      IF (NPRT(14).NE.0) WRITE (6,41) TIME,M,P,AREA,FT,TQM         WINDY
   41 FORMAT(' WIND FORCE',F14.6,I6,2F10.3,3X,3F12.5,3X,3F12.5)   WINDY
      GO TO 99                                                     WINDY
C                                                                  WINDY
C     USE GRID TO CALCULATE WIND FORCE                             WINDOP
C            VP - ORIGIN OF WIND                                   WINDOP
C                                                                  WINDOP
15    AREAT=0.0                                                    WINDOP
      DO 16 I=1,3                                                  WINDOP
      TTF(I)=0.0                                                   WINDOP
      TQM(I)=0.0                                                   WINDOP
16    VP(I) = -FT(I)*10000.0                                       WINDOP
      TEMP=FT(1)**2+FT(2)**2+FT(3)**2                              WINDOP
      IF (TEMP.EQ.0.0) GOTO 99                                     WINDOP
      CALL MAT31(D(1,1,M),FT,FF)                                   WINDOP
      TEMP = 0.0                                                   WINDOP
      IF (FT(1).NE.0.0.OR.FT(2).NE.0.0) GOTO 150                   WINDOP
C                                                                  WINDOP
C     CALCULATE DIRECTION COSINE MATRIX FOR VP COORD. SYS.         WINDOP
C                                                                  WINDOP
      DO 140 I=1,3                                                 WINDOP
      DO 140 J=1,3                                                 WINDOP
140   DVP(I,J)=0.0                                                 WINDOP
      DVP(1,2)=1.0                                                 WINDOP
      DVP(2,1)=1.0                                                 WINDOP
      DVP(3,3)=-1.0                                                WINDOP
      GO TO 141                                                    WINDOP
150   CONTINUE                                                     WINDOP
      DO 110 I=1,3                                                 WINDOP
110   TEMP=TEMP+FT(I)*FT(I)                                        WINDOP
      TEMP = DSQRT(TEMP)                                           WINDOP
      XNORM = DSQRT(FT(1)*FT(1)/TEMP**2+FT(2)*FT(2)/TEMP**2)       WINDOP
      DVP(1,1) = FT(2)/(XNORM*TEMP)                                WINDOP
      DVP(1,2) = -FT(1)/(XNORM*TEMP)                               WINDOP
      DVP(1,3) = 0.0                                               WINDOP
      DVP(2,1) = FT(1)*FT(3)/(XNORM*TEMP*TEMP)                     WINDOP
      DVP(2,2) = FT(2)*FT(3)/(XNORM*TEMP*TEMP)                     WINDOP
      DVP(2,3) = -XNORM                                            WINDOP
      DO 130 I=1,3                                                 WINDOP
130   DVP(3,I) = FT(I)/TEMP                                        WINDOP
141   CONTINUE                                                     WINDOP
      MOELP = MWSEG(7,M)                                           WINDOP
C                                                                  WINDOP
C     PROJECT MM ELLIPSOID UNTO VP-PLANE                           WINDOP
C          AS - PROJECTED ELLIPSE MATRIX                           WINDOP
C                                                                  WINDOP
      CALL DOTT33(D(1,1,M),DVP,DD)                                 WINDOP
      CALL MAT33(BD(7,MM),DD,DDD)                                  WINDOP
      CALL DOT33(D(1,1,M),DDD,DD)                                  WINDOP
```

```
      CALL MAT33(DVP,DD,AM)                                         WINDOP
      DO 101 K=1,3                                                  WINDOP
101   SS(K)=SEGLP(K,M)+BD(K+3,MM)-VP(K)                             WINDOP
      CALL MAT31(DVP,SS,SM)                                         WINDOP
      DO 114 K=1,3                                                  WINDOP
      IF (DABS(SM(K)).LT.EPS(5)) SM(K)=DSIGN(EPS(5),SM(K))          WINDOP
114   CONTINUE                                                      WINDOP
      CALL SOLVR(AM(1,1),AM(2,1),AM(3,1),AM(1,3),AM(2,3),AM(3,3),   WINDOP
     *           AM(1,1),AM(1,3),SM,R(1,1),R(3,1))                  WINDOP
      CALL SOLVR(AM(1,2),AM(2,2),AM(3,2),AM(1,3),AM(2,3),AM(3,3),   WINDOP
     *           AM(2,2),AM(2,3),SM,R(2,2),R(3,2))                  WINDOP
      CALL SOLVR(AM(1,1)+AM(1,2),AM(2,1)+AM(2,2),AM(3,1)+AM(3,2),   WINDOP
     *           AM(1,3),AM(2,3),AM(3,3),AM(1,1)+2.0*AM(1,2)+AM(2,2), WINDOP
     *           AM(1,3)+AM(2,3),SM,R(1,3),R(3,3))                  WINDOP
      R(2,1)=0.0                                                    WINDOP
      R(1,2)=0.0                                                    WINDOP
      R(2,3)=R(1,3)                                                 WINDOP
      DO 102 K=1,3                                                  WINDOP
      DO 102 J=1,2                                                  WINDOP
102   R2(J,K)=R(J,K)                                                WINDOP
      CALL SOLVA(R2,AS(1),AS(2),AS(3))                              WINDOP
C                                                                   WINDOP
C     GET MAJOR & MINOR AXES OF PROJECTED ELLIPSE                   WINDOP
C                                                                   WINDOP
      TEMP=(AS(1)+AS(2))**2-4.0*(AS(1)*AS(2)-AS(3)**2)              WINDOP
      IF (TEMP.LT.0.0) TEMP=0.0                                     WINDOP
      TEMP = DSQRT(TEMP)                                            WINDOP
      AMDA1=(AS(1)+AS(2)+TEMP)/2.0                                  WINDOP
      AMDA2=(AS(1)+AS(2)-TEMP)/2.0                                  WINDOP
      R2(1,1)=AS(3)                                                 WINDOP
      R2(2,1)=AMDA1-AS(1)                                           WINDOP
      R2(1,2)=AMDA2-AS(2)                                           WINDOP
      R2(2,2)=AS(3)                                                 WINDOP
      AMDA1=DABS(AMDA1)                                             WINDOP
      AMDA2=DABS(AMDA2)                                             WINDOP
      B1=DSQRT(1.0/(AMDA1*(R2(1,1)**2+R2(1,2)**2)))                 WINDOP
      B2=DSQRT(1.0/(AMDA2*(R2(2,1)**2+R2(2,2)**2)))                 WINDOP
      R2(1,1)=R2(1,1)*B1                                            WINDOP
      R2(1,2)=R2(1,2)*B2                                            WINDOP
      R2(2,1)=R2(2,1)*B1                                            WINDOP
      R2(2,2)=R2(2,2)*B2                                            WINDOP
C                                                                   WINDOP
C     GET BLOCKING ELLIPSOIDS IN VP COORD. SYS.                     WINDOP
C                                                                   WINDOP
      DO 103 MI=1,MOELP                                             WINDOP
      I=MOWSEG(M,MI*2-1)                                            WINDOP
      II=MOWSEG(M,MI*2)                                             WINDOP
      CALL DOTT33(D(1,1,I),DVP,DD)                                  WINDOP
      CALL MAT33(BD(7,II),DD,DDD)                                   WINDOP
      CALL DOT33(D(1,1,I),DDD,DD)                                   WINDOP
```

```
      CALL MAT33(DVP,DD,AI(1,1,MI))                                     WINDOP
      DO 104 K=1,3                                                      WINDOP
104   SS(K)=SEGLP(K,I)+BD(K+3,II)-VP(K)                                 WINDOP
      CALL MAT31(DVP,SS,SI(1,MI))                                       WINDOP
      DO 115 K=1,3                                                      WINDOP
      IF (DABS(SI(K,MI)).LT.EPS(6)) SI(K,MI)=DSIGN(EPS(6),SI(K,MI))     WINDOP
115   CONTINUE                                                          WINDOP
103   CONTINUE                                                          WINDOP
C                                                                       WINDOP
C     SET-UP GRID AND CHECK EACH RECTANGLE CENTER POINT                 WINDOP
C                                                                       WINDOP
      AREA=DSQRT((R2(1,1)**2+R2(2,1)**2)*(R2(1,2)**2+R2(2,2)**2))       WINDOP
      AREA=AREA/NSTEPS**2                                               WINDOP
      IN=2*NSTEPS+1                                                     WINDOP
      DO 105 I=1,IN                                                     WINDOP
      RXC=R2(1,1)-R2(1,1)*(I-1)/NSTEPS                                  WINDOP
      RYC=R2(2,1)-R2(2,1)*(I-1)/NSTEPS                                  WINDOP
      DO 106 J=1,IN                                                     WINDOP
      RM(1)=(RXC-R2(1,2)*(NSTEPS-J+1)/NSTEPS)*0.9999                    WINDOP
      RM(2)=(RYC-R2(2,2)*(NSTEPS-J+1)/NSTEPS)*0.9999                    WINDOP
      TM(1)=AM(3,3)                                                     WINDOP
      TM(2)=2.0*(RM(1)*AM(1,3)+RM(2)*AM(2,3))                           WINDOP
      TM(3)=RM(1)**2*AM(1,1)+RM(2)**2*AM(2,2)+2.0*RM(1)*RM(2)*AM(1,2)-1.WINDOP
      TEMP=TM(2)**2-4.0*TM(1)*TM(3)                                     WINDOP
      IF (TEMP.LT.0.0) GOTO 106                                         WINDOP
      B1=(DSQRT(TEMP)-TM(2))/(2.0*TM(1))                                WINDOP
      B2=-(DSQRT(TEMP)+TM(2))/(2.0*TM(1))                               WINDOP
      RM(3)=B1                                                          WINDOP
      IF (B2.LT.B1) RM(3)=B2                                            WINDOP
      SN1(1)=RM(1)+SM(1)                                                WINDOP
      SN1(2)=RM(2)+SM(2)                                                WINDOP
      SN1(3)=RM(3)+SM(3)                                                WINDOP
      CALL DOT31(DVP,SN1,XMM)                                           WINDOP
C                                                                       WINDOP
C     CHECK FOR PENETRATION                                             WINDOP
C                                                                       WINDOP
      DO 107 K=1,3                                                      WINDOP
107   XMN(K)=VP(K)-SEGLP(K,N)+XMM(K)                                    WINDOP
      CALL MAT31(D(1,1,N),XMN,XMM)                                      WINDOP
      BET=PL(4,NN)                                                      WINDOP
      BTS=PL(1,NN)*XMM(1)+PL(2,NN)*XMM(2)+PL(3,NN)*XMM(3)               WINDOP
      IF (BTS.GT.BET) GOTO 106                                          WINDOP
C                                                                       WINDOP
C     CHECK FOR BLOCKING ELLIPSOIDS                                     WINDOP
C                                                                       WINDOP
      DO 109 IM=1,MOELP                                                 WINDOP
      X=SN1(1)-SI(1,IM)                                                 WINDOP
      Y=SN1(2)-SI(2,IM)                                                 WINDOP
      TM(1)=AI(3,3,IM)                                                  WINDOP
      TM(2)=2.0*(AI(1,3,IM)*X+AI(2,3,IM)*Y)                             WINDOP
```

```
      TM(3)=AI(1,1,IM)*X**2+AI(2,2,IM)*Y**2+2.0*AI(1,2,IM)*X*Y-1.0       WINDOP
      TEMP=TM(2)**2-4.0*TM(1)*TM(3)                                       WINDOP
      IF (TEMP.LT.0.0) GOTO 109                                           WINDOP
      B1=(-TM(2)+DSQRT(TEMP))/(2.0*TM(1))                                 WINDOP
      B2=(-TM(2)-DSQRT(TEMP))/(2.0*TM(1))                                 WINDOP
      IF (B2.LT.B1) B1=B2                                                 WINDOP
      SNZ=B1+SI(3,IM)                                                     WINDOP
      IF (SNZ.LT.SN1(3)) GOTO 106                                         WINDOP
109   CONTINUE                                                            WINDOP
      CALL DOT31(DVP,RM,SS)                                               WINDOP
      CALL MAT31(D(1,1,M),SS,RM)                                          WINDOP
C                                                                         WINDOP
C     SUM FORCES & TORQUES                                                WINDOP
C                                                                         WINDOP
      AREAT=AREAT+AREA                                                    WINDOP
      DO 111 K=1,3                                                        WINDOP
      TTF(K)=FT(K)*AREA+TTF(K)                                            WINDOP
      RM(K)=RM(K)+BD(K+3,MM)                                              WINDOP
111   FFT(K)=FF(K)*AREA                                                   WINDOP
      CALL CROSS(RM,FFT,TM)                                               WINDOP
      DO 112 K=1,3                                                        WINDOP
112   TQM(K)=TQM(K)+TM(K)                                                 WINDOP
106   CONTINUE                                                            WINDOP
105   CONTINUE                                                            WINDOP
C                                                                         WINDOP
C     ADD FORCE & TORQUE TO U1 & U2 ARRAYS FOR SEGMENT M                   WINDOP
C                                                                         WINDOP
      IF (NPRT(14).NE.0) WRITE(6,200) TIME,M,AREAT,TTF,TQM                WINDOP
200   FORMAT(' WIND FORCE',F14.6,I6,13X,F10.3,3F12.5,3X,3F12.5)           WINDOP
      DO 113 I=1,3                                                        WINDOP
      WF(I,M)=TTF(I)                                                      WINDOP
      U1(I,M)=U1(I,M)+TTF(I)                                              WINDOP
113   U2(I,M)=U2(I,M)+TQM(I)                                              WINDOP
      GO TO 99                                                            WINDOP
C                                                                         WINDOP
C     M = 0: CALCULATE FORCE FUNCTIONS.                                   WINDOP
C                                                                         WINDY
   50 NFORCE = NFVSEG(6)                                                  WINDY
      DO 60 J=1,NFORCE                                                    WINDY
      NFS = IABS(NFVSEG(J))                                               WINDY
      NFT = IABS(NFVNT(J))                                                WINDY
      KFT = NTI(NFT)                                                      WINDY
      FRCE = EVALFD(TIME,KFT,1)                                           WINDY
      IF  (NFVSEG(J).GT.0)  GO TO 52                                      WINDY
      DO  51  I=1,3                                                       WINDY
   51 U2(I,NFS) = U2(I,NFS) + FRCE*QFU(I,J)                               WINDY
      GO TO 60                                                            WINDY
   52 CALL DOT31 (D(1,1,NFS),QFU(1,J),TM)                                 WINDY
      DO  53  I=1,3                                                       WINDY
      U1(I,NFS) = U1(I,NFS) + FRCE*TM(I)                                  WINDY
```

```
53 U2(I,NFS) = U2(I,NFS) + FRCE*QFV(I,J)                          WINDY
60 CONTINUE                                                        WINDY
99 CALL ELTIME (2,37)                                              WINDY
   RETURN                                                          WINDY
   END                                                             WINDY
```

```
      DOUBLE PRECISION FUNCTION XDY(X,D,Y)                              XDY
C                                                     REV IV   07/23/86JTF786
C     FUNCTION ROUTINE TO COMPUTE X.DY OR Y.D'X                         XDY
C                                                                       XDY
      IMPLICIT REAL*8(A-H,O-Z)                                          XDY
      DIMENSION X(3),D(3,3),Y(3)                                        XDY
      XDY = 0.0                                                         XDY
      DO 10 I=1,3                                                       XDY
   10 XDY = XDY + X(I)*(D(I,1)*Y(1)+D(I,2)*Y(2)+D(I,3)*Y(3))            JTF786
      RETURN                                                            XDY
      END                                                               XDY
```

```
      SUBROUTINE YPRDEG(D,A)                                            YPRDEG
C                                                 REV IV    11/26/86YPRFIX
C     COMPUTES YAW PITCH AND ROLL IN DEGREES AND PLACES THEM            YPRDEG
C        INTO THE A ARRAY FOR A GIVEN DIRECTION COSINE MATRIX D.        YPRDEG
C                                                                       YPRDEG
C        ASSUMES D = D(R)D(P)D(Y) , WHERE                               YPRDEG
C                                                                       YPRDEG
C              1  0  0           CP 0 -SP             CY SY 0           YPRDEG
C     D(R) = 0  CR SR  ,D(P) = 0  1  0  AND D(Y) = -SY CY 0           YPRDEG
C            0 -SR CR            SP 0  CP              0  0  1           YPRDEG
C                                                                       YPRDEG
      IMPLICIT REAL*8(A-H,O-Z)                                          YPRDEG
      DIMENSION A(3),D(3,3)                                             YPRDEG
      COMMON/CNSNTS/ PI,RADIAN,G,THIRD,EPS(24),                         YPRDEG
     *               UNITL,UNITM,UNITT,GRAVTY(3),TWOPI                  TWOPI
      IF (DABS(D(1,1)).LE.EPS(15).AND.DABS(D(1,2)).LE.EPS(15))GOTO10    YPRFIX
      IF (DABS(D(2,3)).LE.EPS(15).AND.DABS(D(3,3)).LE.EPS(15))GOTO10    YPRFIX
      YAW = DATAN2(D(1,2),D(1,1))                                       YPRDEG
      ROLL = DATAN2(D(2,3),D(3,3))                                      YPRDEG
      GO TO 11                                                          YPRDEG
   10 YAW = DATAN2(-D(2,1),D(2,2))                                      YPRDEG
      ROLL = 0.0                                                        YPRDEG
   11 PITCH = -DASIN(D(1,3))                                            YPRDEG
      IF (DABS(ROLL).LE.0.5*PI) GO TO 20                                YPRDEG
      IF (DABS(YAW ).LE.0.5*PI) GO TO 20                                YPRDEG
      PITCH = DSIGN(PI-DABS(PITCH),PITCH)                               YPRDEG
      YAW  = DATAN2(-D(1,2),-D(1,1))                                    YPRDEG
      ROLL = DATAN2(-D(2,3),-D(3,3))                                    YPRDEG
   20 A(1) = YAW/RADIAN                                                 YPRDEG
      A(2) = PITCH/RADIAN                                               YPRDEG
      A(3) = ROLL/RADIAN                                                YPRDEG
      RETURN                                                            YPRDEG
      END                                                               YPRDEG
```

REFERENCES

1. Fleck, J.T., Butler, F.E. and Vogel, SD.L., "An Improved Three Dimensional Computer Simulation of Crash Victims," NHTSA Report Nos. DOT-HS-801-507 through 510, April 1975, Vols. 1-4.

2. Fleck, J.T., Butler, F.E., "Development of an Improved Computer Model of the Human Body and Extremity Dynamics," Report No. AMRL-TR-75-14, July 1975 (NTIS No. AD-A014 816).

3. Butler, F.E. and Fleck, J.T., "Advanced Restraint System Modeling," Report No. AFAMRL-TR-80-14, May 1980 (NTIS No. AD-A088 029).

4. Fleck, J.T., Butler, F.E., and DeLeys, N.J., "Validation of the Crash Victim Simulator," Calspan Report Nos. ZS-5881-V-1 through 4, DOT-HS-806-279 through 282, 1982, Vols 1-4 (NTIS No. PC E99, PB 86-212420).

5. Butler, F.E., Fleck, J.T. and Difranco, D.A., "Modeling of Whole-Body Response to Windblast," Report No. AFAMRL-TR-83-073, October 1983 (NTIS No. AD-B079 184).

6. Leetch, B.D., Bowman, W.L., "Articulated Total Body (ATB) VIEW Program Software Report," Report Nos. AMRL-TR-81-111, Vols 1 & 2, June 1983 (NTIS Nos. AD-B075 161 & 162).

*U.S.Government Printing Office: 1988 — 548-053/80135

END
DATE
10-88
DTIC